Report of Final Project

Name : R. Shalini

Batch Code : TN-DA-FNB03

Contact No : 9080585426

E-Mail : shaalu2408@gmail.com

Domain : E-Commerce

Project Title :

“Data Cleaning, EDA and Visualization of Customer Analytics Loyalty Vs Fraud in E-Commerce using Python and Power BI Interactive Dashboards”

Submission Date : 9.12.2025

Mentor : Mr. Kumaran

Raw Dataset Link :

https://drive.google.com/file/d/1DD0-GK7AQ2pkwau0S61Z9P7VQ2l5AMz9/view?usp=sharing

Cleaned Dataset Link :

https://drive.google.com/file/d/1H0PSGpHyCxRUFz-o7NMVhkWJhabGwyWO/view?usp=drive_link

Index

Project Title:

Data Cleaning, EDA and Visualization of Customer Analytics Loyalty Vs Fraud in E-Commerce using Python and Power BI Interactive Dashboards

---

Domain:

The Domain of the project is **E-commerce**

Objective:

1. To analyse global e-commerce customer data to identify loyal customers.
2. To detect fraudulent behaviours that affects sales and profit.
3. To improve customer retention.
4. To reduce fraud losses and to enhance decision-making.
5. To use data analytics and visualization for better understanding.

Outcome:

Built analytics dashboard between loyal vs fraudulent customers and to enhanced fraud detection accuracy and improved understanding of loyal customer behaviour. Supported strategic actions for customer retention and fraud prevention.

Dataset Information:

Source:

Kaggle

Year / Timeline:

Data collected during June 2020 to June 2025

Dataset Description:

1. **Customer_id** - Gives information about customers.
2. **Age** - Age of each customer.
3. **Gender** - Sex of each customer.
4. **Country** - Location of the customers.
5. **Avg_order_value** - Average amount of money spent.
6. **Total_orders** - Total number of orders.
7. **Last_purchase** - Customer recently (last) purchased amount.
8. **Is_fraudulent** - Customer is fraudulent '1' or '0' not fraud.
9. **Preferred_category** - Product depends on which category.
10. **Email_open_rate** - Rate of customers who opened Email.
11. **Customer_since** - Date of joining of customers.
12. **Loyalty_score** - Score of each customer based on loyalty.
13. **Churn_risk** - Customers stop buying product.

Type of Analysis:

Descriptive Analysis (What has happened or past information)

- Analyse total sales, number of customers and their purchasing details.
- Identify loyalty patterns, customers who purchased frequently.

Diagnostic Analysis (why happened or root cause of problems)

- Compare purchased and return patters between loyal vs fraud customers.
- Study payment method correlations with fraud rates.

Predictive Analysis (What is likely to happen or analysing problem)

- Probability of customer being loyal.
- Probability of transaction being fraudulent.

Prescriptive Analysis (for future improvement or recommendations for business decisions)

- Recommend strategies to retain loyal customers.
- Suggest fraud prevention actions.

Stages for DA Project

Stage 1 – Problem Definition and Dataset Selection

Define the business problem and expected outcome

- Business problem:
    1. Identifying Loyal Customers - Customers frequently purchase, engage with promotions and generate consistent revenue.
    2. Detecting Fraudulent transactions - fake accounts, false returns or payment fraud and reputational damage.
- Expected outcome:
    1. Built analytics dashboard between loyal vs fraudulent customers and to enhanced fraud detection.
    2. Supported strategic actions for customer retention and fraud prevention.

Choose dataset and explain its source, size, and features

- **Dataset:** Data Cleaning, EDA and Visualization of Customer Analytics Loyalty Vs Fraud in E-Commerce using Python and Power BI Interactive Dashboards
- **Sources:** Kaggle.
- **Size:**
- * Rows - 5000
  * Columns - 13
- **Features:** Customer_id, Age, Gender, Country, Avg_order_value, Total_orders, Last_purchase, Is_fraudulent, Preferred_category, Email_open_rate, Customer_since, Loyalty_score, Churn_risk.

Import libraries, load dataset:

code:

```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
df= pd.read_csv('/content/synthetic_ecommerce_churn_dataset.csv')
df
```

o/p:

df

	customer_id	age	gender	country	avg_order_value	total_orders	last_purchase	is_fraudulent	preferred_category	email_open_
0	CUST_8270	30	Female	Brazil	101.08	8	176	1	Beauty	
1	CUST_1860	53	Female	USA	90.39	10	88	0	Electronics	
2	CUST_6390	73	Male	Australia	83.28	6	203	0	Sports	
3	CUST_6191	30	Other	Japan	109.90	9	346	1	Electronics	
4	CUST_6734	29	Female	Canada	269.38	16	342	0	Fashion	
...	...	...	...	...	...	...	...	...	...	
4995	CUST_8533	49	Female	UK	132.16	9	306	0	Electronics	
4996	CUST_5616	23	Female	UK	47.81	10	296	0	Home	
4997	CUST_2140	79	Male	Japan	224.97	16	84	0	Beauty	
4998	CUST_6730	62	Male	USA	220.33	8	254	0	Fashion	
4999	CUST_8465	38	Male	Canada	75.57	6	326	0	Home	

5000 rows × 13 columns

Dataset description (rows, columns, features):

1.Rows and columns:

Code**:**

df.shape

o/p:

```
df.shape
(5000, 13)
```

2.Features:

Code:

df.columns

o/p:

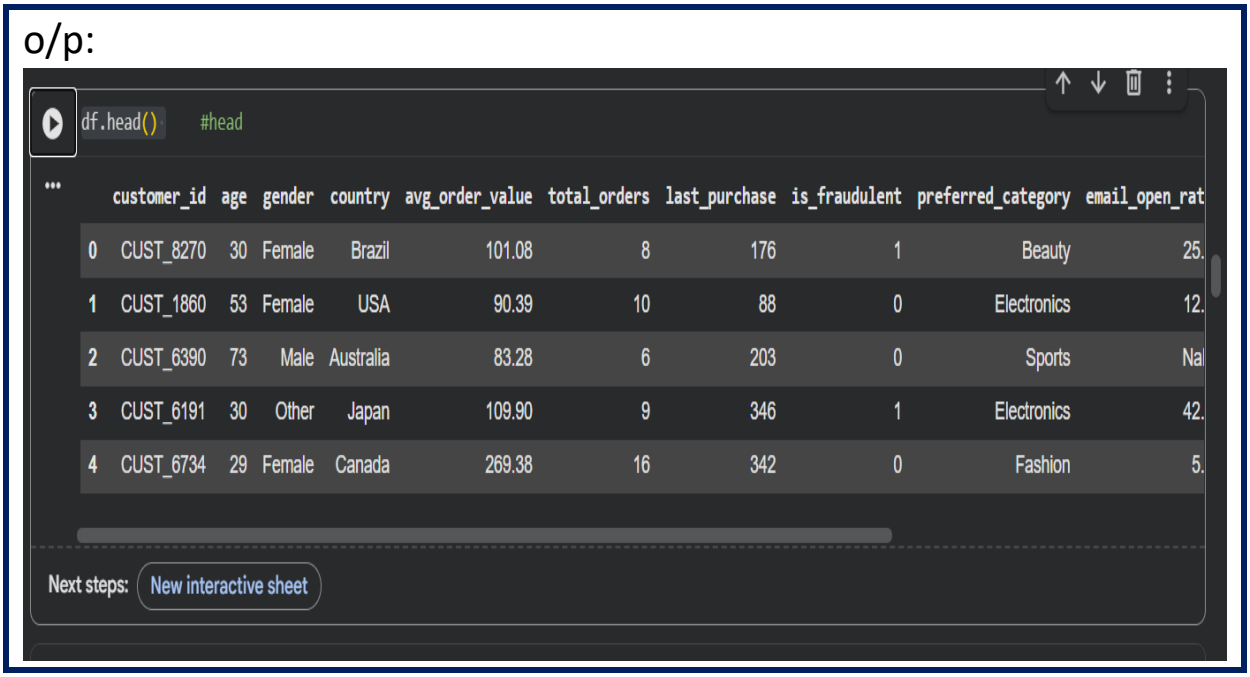
```
df.columns       # all Column names

Index(['customer_id', 'age', 'gender', 'country', 'avg_order_value',
       'total_orders', 'last_purchase', 'is_fraudulent', 'preferred_category',
       'email_open_rate', 'customer_since', 'loyalty_score', 'churn_risk'],
      dtype='object')
```

Initial EDA (head, info, describe, shape, null checks):

1.Head = First five rows

Code: **df.head()**

o/p:



```
df.head()    #head
```

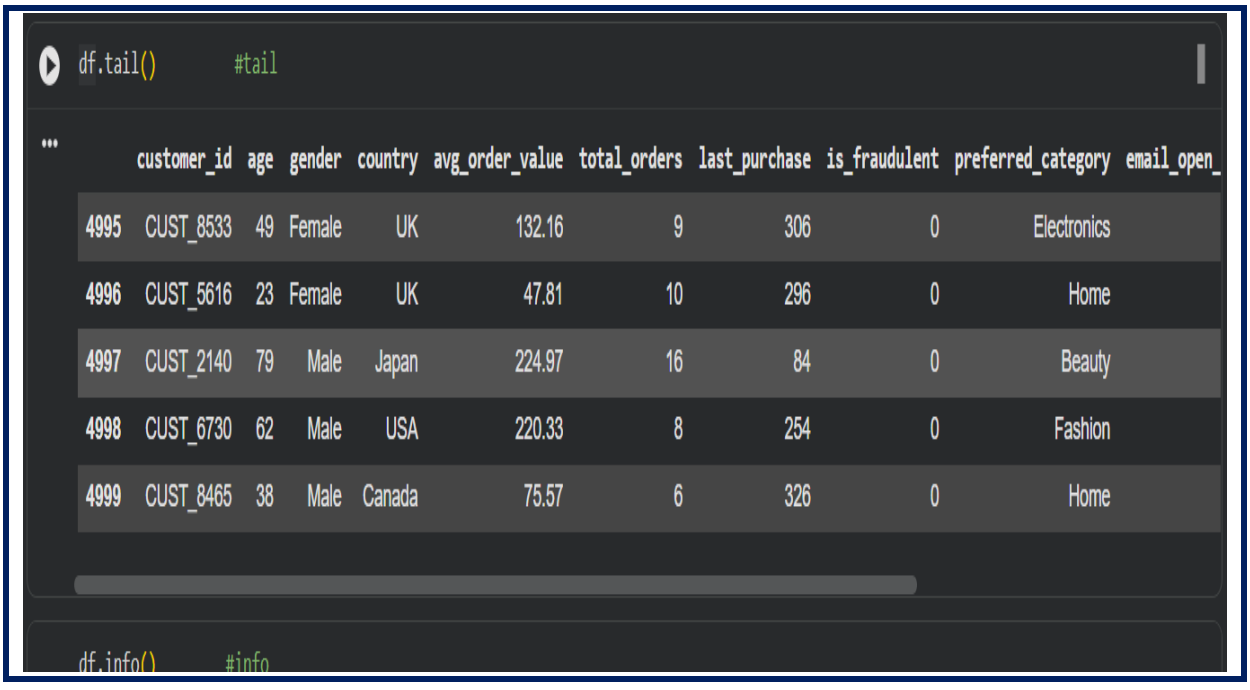
	customer_id	age	gender	country	avg_order_value	total_orders	last_purchase	is_fraudulent	preferred_category	email_open_rat
0	CUST_8270	30	Female	Brazil	101.08	8	176	1	Beauty	25.
1	CUST_1860	53	Female	USA	90.39	10	88	0	Electronics	12.
2	CUST_6390	73	Male	Australia	83.28	6	203	0	Sports	Na
3	CUST_6191	30	Other	Japan	109.90	9	346	1	Electronics	42.
4	CUST_6734	29	Female	Canada	269.38	16	342	0	Fashion	5.

Next steps: New interactive sheet

2.Tail = Last five rows

Code: **df.tail()**

o/p:



```
df.tail()    #tail
```

	customer_id	age	gender	country	avg_order_value	total_orders	last_purchase	is_fraudulent	preferred_category	email_open_
4995	CUST_8533	49	Female	UK	132.16	9	306	0	Electronics	
4996	CUST_5616	23	Female	UK	47.81	10	296	0	Home	
4997	CUST_2140	79	Male	Japan	224.97	16	84	0	Beauty	
4998	CUST_6730	62	Male	USA	220.33	8	254	0	Fashion	
4999	CUST_8465	38	Male	Canada	75.57	6	326	0	Home	

```
df.info()    #info
```

3.Information = Tells total number of columns, Null values, Data type

Code: **df.info()**

o/p

```
df.info()        #info

<class 'pandas.core.frame.DataFrame'>
RangeIndex: 5000 entries, 0 to 4999
Data columns (total 13 columns):
 #   Column              Non-Null Count  Dtype
---  ------              --------------  -----
 0   customer_id         5000 non-null   object
 1   age                 5000 non-null   int64
 2   gender              5000 non-null   object
 3   country             5000 non-null   object
 4   avg_order_value     4750 non-null   float64
 5   total_orders        5000 non-null   int64
 6   last_purchase       5000 non-null   int64
 7   is_fraudulent       5000 non-null   int64
 8   preferred_category  5000 non-null   object
 9   email_open_rate     4750 non-null   float64
 10  customer_since      5000 non-null   object
 11  loyalty_score       5000 non-null   int64
 12  churn_risk          5000 non-null   float64
dtypes: float64(3), int64(5), object(5)
memory usage: 507.9+ KB
```

4.Numerical Columns = Shows all numerical column

Code: **df.select_dtypes(include=['int64','float64']).columns**

o/p:

```
df.select_dtypes(include=['int64','float64']).columns      # Numerical columns

Index(['Age', 'Avgorder_value', 'Total_orders', 'Is_fraudlent',
       'Email_open_rate', 'Loyalty_score', 'churn_risk', 'Revenue'],
      dtype='object')
```

5.Categorical Column = shows all category columns

Code: **df.select_dtypes(include=['object', 'category']).columns**

o/p:

```
df.select_dtypes(include=['object','category']).columns     # categorical columns

Index(['customer_id', 'gender', 'country', 'preferred_category',
       'customer_since'],
      dtype='object')
```

6.Satistical Column = Tells count, mean, minimum, maximum, std, 25%,50%,75% values of all numerical values.

Code**: df.describe()**

o/p:

```
df.describe()          #describe
```

	age	avg_order_value	total_orders	last_purchase	is_fraudulent	email_open_rate	loyalty_score	churn_risk
count	5000.000000	4750.000000	5000.000000	5000.000000	5000.000000	4750.000000	5000.000000	5000.000000
mean	48.163200	108.442857	10.027000	180.073200	0.025800	50.714842	50.039400	0.284484
std	17.880797	69.265559	3.163838	104.926518	0.158554	29.098706	28.832151	0.159690
min	18.000000	10.660000	0.000000	0.000000	0.000000	0.000000	1.000000	0.000000
25%	33.000000	57.805000	8.000000	89.000000	0.000000	25.225000	25.000000	0.160000
50%	48.000000	93.190000	10.000000	178.000000	0.000000	50.950000	50.000000	0.260000
75%	64.000000	142.197500	12.000000	270.000000	0.000000	76.800000	75.000000	0.390000
max	79.000000	555.460000	23.000000	364.000000	1.000000	100.000000	99.000000	0.900000

7.Sample = Randomly choose any 10 rows

Code: **df.sample(10)**

o/p

```
df.sample(10)        # any 10 sample of customer in dataset
```

	customer_id	age	gender	country	avg_order_value	total_orders	last_purchase	is_fraudulent	preferred_category	email_open_
4791	CUST_7019	42	Male	Japan	61.71	7	54	0	Fashion	
2348	CUST_7788	31	Male	Australia	57.17	8	73	0	Home	
1455	CUST_7368	23	Female	Australia	87.75	11	11	0	Beauty	
3259	CUST_5173	33	Female	Germany	84.46	6	310	0	Electronics	
2971	CUST_3222	46	Male	Germany	39.81	13	285	0	Beauty	
3271	CUST_3240	50	Male	France	116.54	11	288	0	Home	
1846	CUST_6548	25	Female	UK	99.75	15	284	0	Electronics	
2433	CUST_3990	42	Male	India	76.31	9	67	0	Fashion	
461	CUST_1728	38	Male	UK	51.64	10	278	0	Fashion	
3152	CUST_7696	61	Female	Canada	46.60	11	299	1	Beauty	

8. Unique values = Number of unique values in each column

Code: df.nunique()

o/p:

```
df.nunique()          # unique values in numerical columns
```

	0
customer_id	3809
age	62
gender	3
country	10
avg_order_value	4221
total_orders	24
last_purchase	365
is_fraudulent	2
preferred_category	5
email_open_rate	989
customer_since	1701
loyalty_score	99
churn_risk	88

dtype: int64

9.Null value = Checking null values
Code: df.isnull().sum()
o/p:

	0
customer_id	0
age	0
gender	0
country	0
avg_order_value	250
total_orders	0
last_purchase	0
is_fraudulent	0
preferred_category	0
email_open_rate	250
customer_since	0
loyalty_score	0
churn_risk	0

dtype: int64

10.Index = Showing start, stop and step position
Code: df.index
o/p:

```
df.index          # index of the dataset
```

```
RangeIndex(start=0, stop=5000, step=1)
```

Stage 2 – Data Cleaning and Pre-processing

Handle missing values (impute or drop):

Here missing values are imputed by fillna method by using median.

Code:

```
df['avg_order_value']=df['avg_order_value'].fillna(df['avg_order_value'].median())
df['email_open_rate']=df['email_open_rate'].fillna(df['email_open_rate'].median())
```

Code: **df.isnull().sum()**

Before cleaning

```
df.isnull().sum()  # che
...
0
customer_id 0
age 0
gender 0
country 0
avg_order_value 250
total_orders 0
last_purchase 0
is_fraudulent 0
preferred_category 0
email_open_rate 250
customer_since 0
loyalty_score 0
churn_risk 0
dtype: int64
```

After cleaning

```
df.isnull().sum()
0
customer_id 0
age 0
gender 0
country 0
avg_order_value 0
total_orders 0
last_purchase 0
is_fraudulent 0
preferred_category 0
email_open_rate 0
customer_since 0
loyalty_score 0
churn_risk 0
```

Handle duplicates:

Code: df.duplicated().sum()

o/p:

```
df.duplicated().sum()

np.int64(0)
```

Replacing headers names correctly:

Capitalise first letter of all headers.

Before Replace

Code: **df.columns**

o/p:

```
df.columns       # before replace

Index(['customer_id', 'age', 'gender', 'country', 'avg_order_value',
       'total_orders', 'last_purchase', 'is_fraudulent', 'preferred_category',
       'email_open_rate', 'customer_since', 'loyalty_score', 'churn_risk'],
      dtype='object')
```

by using **rename** method,

code:

```
df.rename(columns={'customer_id':'CustomerID','age':'Age','gender':'Gender','country':'Country',
                   'avg_order_value':'Avgorder_value','total_orders':'Total_orders','last_purchase':
                   'Last_purchase','is_fraudulent':'Is_fraudlent','preferred_category':'Preferred_category',
                   'email_open_rate':'Email_open_rate','customer_since':'Customer_since',
                   'loyalty_score':'Loyalty_score','chrun_risk':'Chrun_risk'},inplace=True)
```

After Replace

Code: **df.columns**

o/p:

```
df.columns      # after replace

Index(['CustomerID', 'Age', 'Gender', 'Country', 'Avgorder_value',
       'Total_orders', 'Last_purchase', 'Is_fraudlent', 'Preferred_category',
       'Email_open_rate', 'Customer_since', 'Loyalty_score', 'churn_risk'],
      dtype='object')
```

Silicing for 10 to 20 rows in 'Country' column:

Code: **df.Country[10:20]**

o/p:

```
df.Country[10:20]

      Country
10         UK
11         UK
12     Brazil
13      India
14     France
15      India
16      India
17     France
18      China
19     Canada

dtype: object
```

Integer indexing by row and column:

Code: **df.iloc[5:15,8:13]**

o/p:

	Preferred_category	Email_open_rate	Customer_since	Loyalty_score	churn_risk
5	Fashion	95.9	2020-12-25	67	0.53
6	Beauty	4.1	2025-05-26	91	0.37
7	Fashion	32.8	2022-01-13	26	0.18
8	Electronics	73.0	2021-02-15	24	0.22
9	Sports	20.4	2023-04-02	52	0.35
10	Home	19.7	2025-06-15	90	0.14
11	Sports	14.3	2023-07-31	53	0.14
12	Home	99.9	2020-12-22	19	0.45
13	Electronics	90.2	2023-07-20	60	0.42
14	Home	13.6	2024-06-16	21	0.30

Label base indexing on 3000th row:

Code: df.loc[3000]

o/p:

```
df.loc[3000]  # shows entire
```

	3000
CustomerID	CUST_9671
Age	57
Gender	Male
Country	UK
Avgorder_value	115.83
Total_orders	10
Last_purchase	208
Is_fraudlent	0
Preferred_category	Fashion
Email_open_rate	8.1
Customer_since	14-04-2021
Loyalty_score	16

Integer position based - Scalar accessor:

Code: df.iat[4235,3]

o/p:

```
df.iat[4235,3]    # shows the 4235th row, 3 rd column value based on integer position

'Germany'
```

Label based - Scalar accessor:

Code: df.at[4235,'Country']

o/p:

```
df.at[4235,'Country']    # shows the 4235th row, Country column value based on label scalar accessor

'Germany'
```

Sorted country column by alphabetically order:

Code:

```
df_sorted = df.sort_values("Country")
df_sorted[['CustomerID','Country']]
```

o/p:

```
df_sorted = df.sort_values("Country")
df_sorted[['CustomerID','Country']]
```

	CustomerID	Country
3721	CUST_4062	Australia
2268	CUST_6854	Australia
1493	CUST_4897	Australia
1494	CUST_5942	Australia
2266	CUST_2827	Australia
...	...	...
430	CUST_8954	USA
428	CUST_9984	USA
3762	CUST_9363	USA
4313	CUST_5752	USA
2302	CUST_2422	USA

4950 rows × 2 columns

Treat outliers if required:

Before removing outliers

Code: df.describe().round(2)

o/p:

	Age	Avgorder_value	Total_orders	Last_purchase	Is_fraudlent	Email_open_rate	Loyalty_score	churn_risk
count	5000.00	5000.00	5000.00	5000.00	5000.00	5000.00	5000.00	5000.00
mean	48.16	107.68	10.03	180.07	0.03	50.73	50.04	0.28
std	17.88	67.59	3.16	104.93	0.16	28.36	28.83	0.16
min	18.00	10.66	0.00	0.00	0.00	0.00	1.00	0.00
25%	33.00	59.56	8.00	89.00	0.00	26.40	25.00	0.16
50%	48.00	93.19	10.00	178.00	0.00	50.95	50.00	0.26
75%	64.00	138.81	12.00	270.00	0.00	75.30	75.00	0.39
max	79.00	555.46	23.00	364.00	1.00	100.00	99.00	0.90

By identifying outliers in each column, we find Avgorder_value column have many outlier rows.

Code:

```
Avgorder_value_threshold=df['Avgorder_value'].quantile(0.99).round(2)
Avgorder_value_threshold
```

o/p:

```
Avgorder_value_threshold=df['Avgorder_value'].quantile(0.99).round(2)   # To find outliers in Avgorder_value
Avgorder_value_threshold

np.float64(336.68)
```

code: df[df['Avgorder_value']>Avgorder_value_threshold]
(rows having outliers in Avgorder_value)

o/p:

```
df[df['Avgorder_value']>Avgorder_value_threshold]  # rows having outliers
```

4719	CUST_7996	57	Male	France	360.12	16
4720	CUST_9712	70	Male	UK	361.66	9
4721	CUST_2401	75	Male	France	361.97	9
4722	CUST_2497	41	Female	USA	362.07	11
4723	CUST_2895	57	Other	Japan	365.38	10
4724	CUST_2230	53	Male	Australia	366.02	7
4725	CUST_8490	39	Male	Brazil	367.38	11
4726	CUST_2409	31	Female	France	375.82	8
4727	CUST_6944	47	Male	USA	380.30	7
4728	CUST_7941	46	Female	Germany	384.14	6
4729	CUST_5117	65	Female	Brazil	389.32	16
4730	CUST_4760	53	Male	France	398.16	10
4731	CUST_6189	41	Female	France	405.36	12
4732	CUST_4957	78	Male	Australia	405.53	17
4733	CUST_8355	49	Female	France	405.77	8
4734	CUST_7980	61	Female	Canada	407.34	11

Code: df=df[df['Avgorder_value'] <= Avgorder_value_threshold]
removing outliers in Avgorder_value column.
o/p:

```
df=df[df['Avgorder_value'] <=  Avgorder_value_threshold]
df[['CustomerID','Avgorder_value']]
```

	CustomerID	Avgorder_value
0	CUST_9340	10.66
1	CUST_8604	10.70
2	CUST_2440	11.48
3	CUST_7530	11.61
4	CUST_2826	12.50
...	...	...
4995	CUST_9728	93.19
4996	CUST_9801	93.19
4997	CUST_9858	93.19
4998	CUST_9970	93.19
4999	CUST_9984	93.19

4950 rows × 2 columns

After removing outliers

Shape = showing number of rows and columns
Code: df.shape
o/p:

```
df.shape  # after removed outliers
```

```
(4950, 13)
```

Statistical Column = Tells count, mean, minimum, maximum, std, 25%,50%,75% values of all numerical values.

Code: df.describe().round(2)
o/p:

```
df.describe().round(2)   # after remover outliers
```

	Age	Avgorder_value	Total_orders	Last_purchase	Is_fraudlent	Email_open_rate	Loyalty_score	churn_risk
count	4950.00	4950.00	4950.00	4950.00	4950.00	4950.00	4950.00	4950.00
mean	48.14	104.79	10.03	180.03	0.03	50.68	50.06	0.28
std	17.91	61.21	3.16	104.81	0.16	28.36	28.85	0.16
min	18.00	10.66	0.00	0.00	0.00	0.00	1.00	0.00
25%	33.00	59.09	8.00	89.00	0.00	26.40	25.00	0.16
50%	48.00	93.19	10.00	178.00	0.00	50.95	50.00	0.26
75%	64.00	137.28	12.00	270.00	0.00	75.28	76.00	0.39
max	79.00	336.67	23.00	364.00	1.00	100.00	99.00	0.90

Now, Avgorder_value value is changed from max 555.46 to max 336.67, its actually **Avgorder_value_threshold**

Check skewness and apply transformations:

```
numerical_columns=df.select_dtypes(include=['int64','float64']).columns
df[numerical_columns].skew().round(2)
```

if skew value >0 = normally distributed,
skew value between 0.5 to 1 = moderate skew,
skew value < 1 = high skewness.

```
df[numerical_columns].skew().round(2)
```

```
...
                      0
Age                0.04
Avgorder_value     1.03
Total_orders       0.33
Last_purchase      0.04
Is_fraudlent       5.95
Email_open_rate   -0.04
Loyalty_score     -0.00
churn_risk         0.65

dtype: float64
```

In this dataset highly skewness is present in "Is_fraudulent" column, but "No transformation" is needed for binary columns (naturally having high skew value).

Convert data types if needed:

Code:

```
from datetime import datetime as dt
from datetime import date, time, timedelta, timezone
import pytz
```

```
df.loc[:,Customer_since’] =pd.to_datetime(df['Customer_since'],
errors='coerce')

today=dt.today().date()
df.loc[:,'Last_purchase']=pd.to_timedelta(df['Last_purchase'],unit='D')
```

Before cleaning datatypes

CustomerID	object
Age	int64
Gender	object
Country	object
Avgorder_value	float64
Total_orders	int64
Last_purchase	int64
Is_fraudlent	int64
Preferred_category	object
Email_open_rate	float64
Customer_since	object
Loyalty_score	int64
churn_risk	float64

dtype: object

After cleaning datatypes

	0
CustomerID	object
Age	int64
Gender	object
Country	object
Avgorder_value	float64
Total_orders	int64
Last_purchase	timedelta64[ns]
Is_fraudlent	int64
Preferred_category	object
Email_open_rate	float64
Customer_since	datetime64[ns]
Loyalty_score	int64
churn_risk	float64

dtype: object

Feature transformations (date parts, derived fields if required for analysis):

1.creating new column and named as Revenue:

Code:

```
df.loc[:,'Revenue']=df['Avgorder_value']*df['Total_orders']
```

2.By using loyalty score, created a new column Loyalty_type (High Loyal, Moderate Loyal, Low Loyal, Fraud)

Code:

```
def Loyalty_score(score):
  if 80 <= score <= 100:
    return "High Loyal"
  elif 50 <= score <= 79:
    return "Moderate Loyal"
  elif 20 <= score <= 49:
    return "Low Loyal"
  else:
    return "Fraud"
df.loc[:,'Loyalty_type']=df['Loyalty_score'].apply(Loyalty_score)
```

3.created new column Loyalty bucket based on loyalty score

Code:

```
df.loc[:,'Loyalty_bucket']= pd.cut(df['Loyalty_score'], bins=[0,40,70,100],
labels =['Low','Medium','High']
```

4.Created new column Loyalty or fraud based on loyalty bucket and is fraudulent
Code:

```
df.loc[:,'Loyalty_or_fraud']=df['Loyalty_bucket'].astype('string')+"|"+
                                df['Is_fraudulent'].map({0:'Not Fraud',1:'Fraud'})
```

5.created new column, by checking weekend days
Code:

```
df.loc[:,'Is_weekend']=df['Customer_since'].dt.day_name().isin(['saturday',
'Sunday'])
```

6.created new column by extracted year from Customer_since
Code:

```
df.loc[:,'Year']=df['Customer_since'].dt.year
```

7.created new column based on churn_risk
Code:

```
risk_label=[]
for value in df['churn_risk']:
  if value <= 0.30:
   risk_label.append("Low Risk")
  elif value <= 0.70:
    risk_label.append("Medium Risk")
  else:
    risk_label.append("High Risk")
df.loc[:,'Risk_level']=risk_label
```

8.created new column based on loyalty score, named as 'loyalty level'.

Code:

```
Loyalty_level = []
for score in df['Loyalty_score']:
  if score <= 40:
    Loyalty_level.append("Low Rate")
  elif score <= 80:
    Loyalty_level.append("Moderate")
  else:
    Loyalty_level.append("High Rate")
df.loc[:,'Loyalty_level']=Loyalty_level
```

9.Created a new column, named as Country loyalty

Code:

```
df.loc[:,'Country_Loyalty']=df['Country'].astype(str)+"_"+
                                                    df['Loyalty_level'].astype(str)
```

10.created a new column as Customer_tag by concatenation of 'Country',' Preferred_category', 'Is_fraudulent'.

Code:

```
df.loc[:,'Customer_tag']=(df['Country']+"|"+df['Preferred_category']+
                                                "|
Fraud:"+df['Is_fraudulent'].astype(str))
```

11.Created a new column as Revenue_loyal by revenue amount.

Code:

```
df.loc[:,'Revenue_loyal']=df['Revenue'].apply
                          (lambda x: 'Highly Loyal'if x > 1500
                            else ('Medium Loyal' if x >= 500
                            else 'Low Loyal'))
```

Now, cleaned dataset is ready

Code: **df**

o/p:

```
df      # cleaned dataset
```

	CustomerID	Age	Gender	Country	Avgorder_value	Total_orders	Last_purchase	Is_fraudlent	Preferred_cate
0	CUST_9340	43	Female	Germany	10.66	13	346 days	0	Electro
1	CUST_8604	44	Male	China	10.70	6	170 days	0	S
2	CUST_2440	77	Male	Canada	11.48	9	245 days	0	Fas
3	CUST_7530	78	Female	China	11.61	11	121 days	0	S
4	CUST_2826	54	Male	China	12.50	10	182 days	0	Be
...	...	...	...	...	...	...	...	...	
4995	CUST_9728	33	Female	Germany	93.19	12	278 days	0	H

---

code: **df.shape**

o/p:

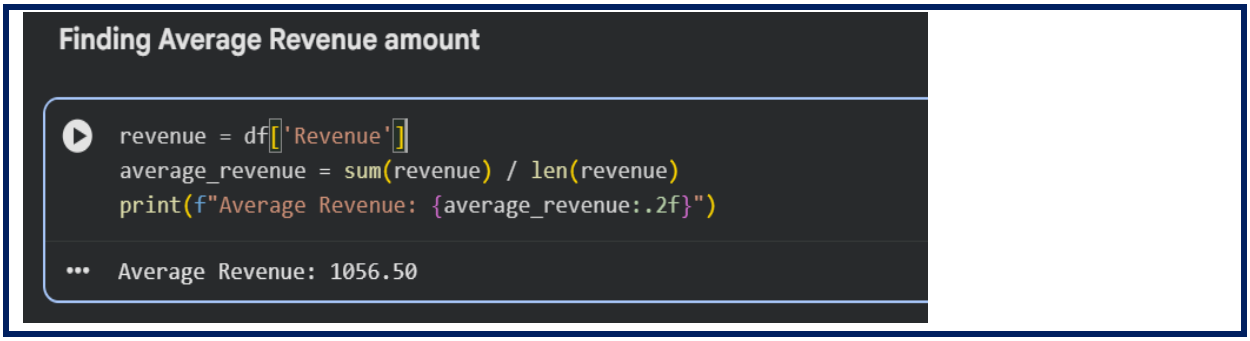
```
df.shape   # cleaned dataset

(4950, 24)
```

Finding Average Revenue amount:

Code:

```
revenue = df['Revenue']
average_revenue = sum(revenue) / len(revenue)
print(f"Average Revenue: {average_revenue:.2f}")
```

o/p:

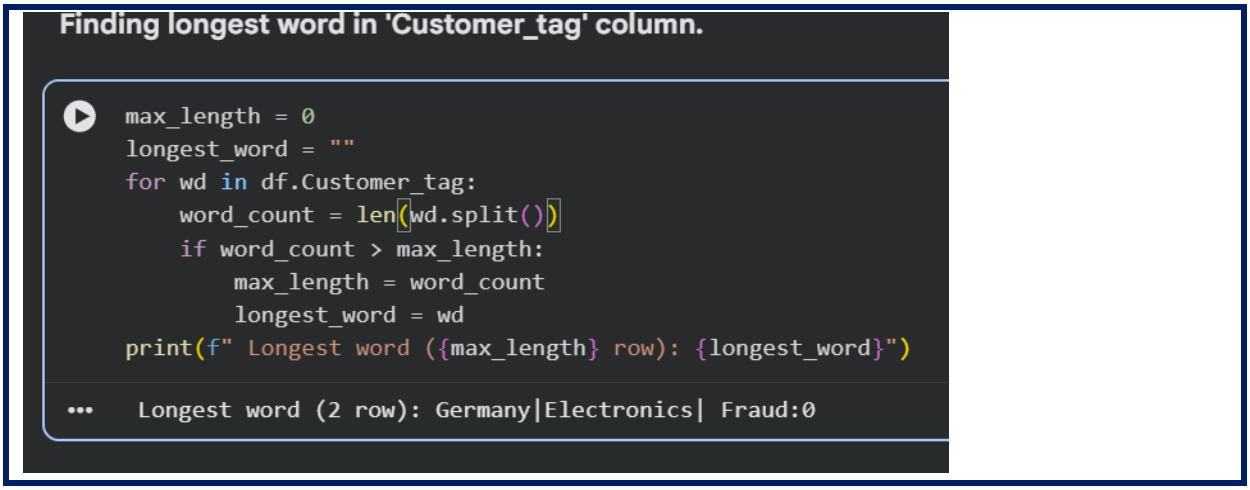


Finding longest word in 'Customer_tag' column:

Code:

```
max_length = 0
longest_word = ""
for wd in df.Customer_tag:
   word_count = len(wd.split())
   if word_count > max_length:
      max_length = word_count
      longest_word = wd
print(f" Longest word ({max_length} row): {longest_word}")
```

o/p:

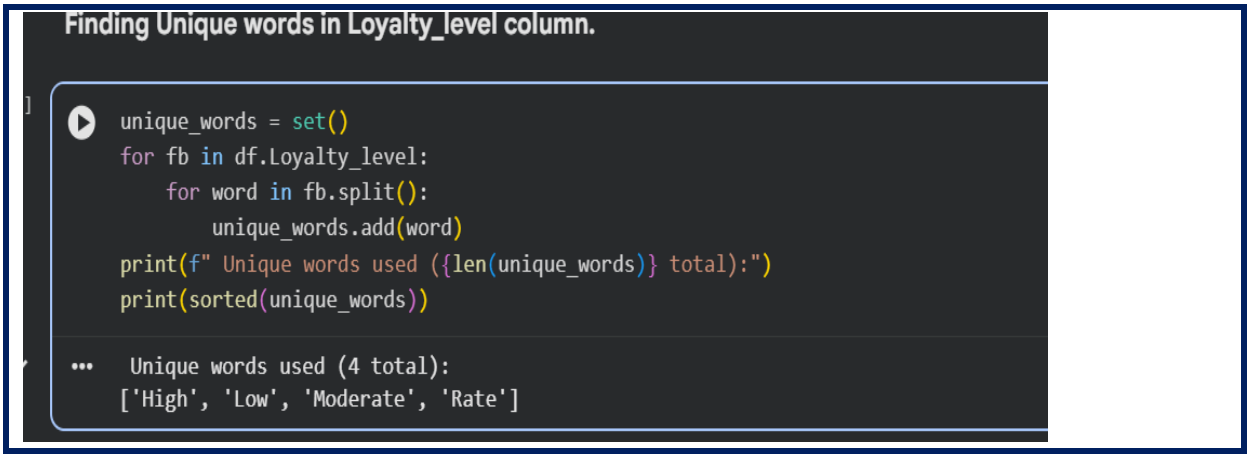


Finding Unique words in Loyalty_level column:

Code:

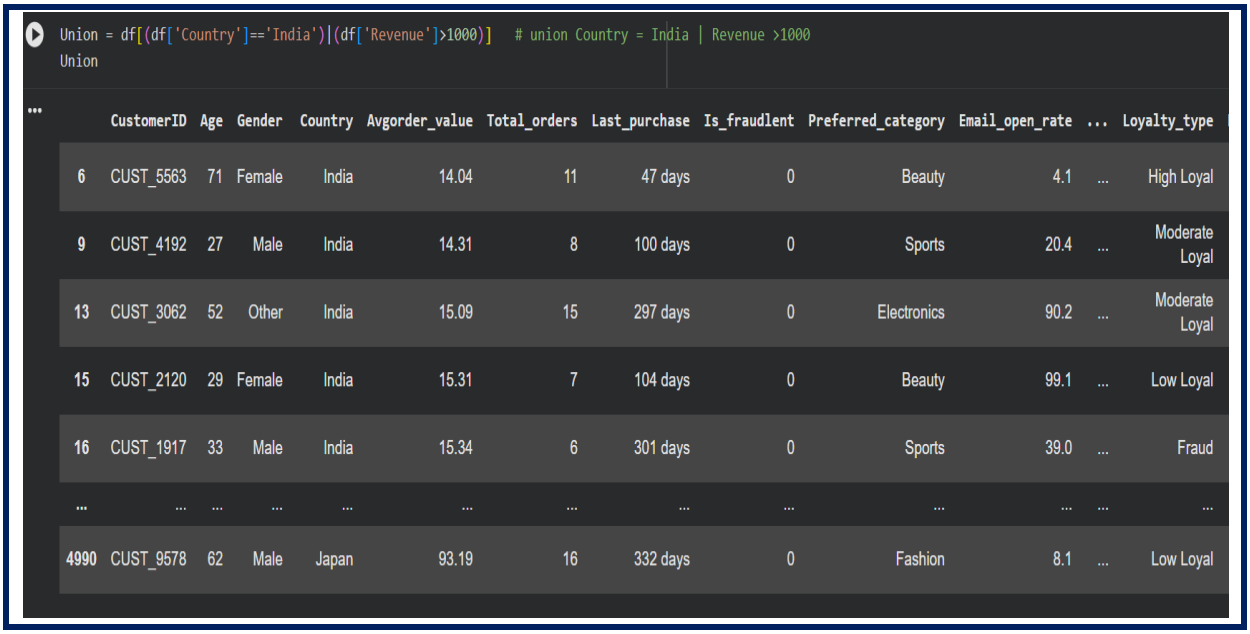
```
unique_words = set()
for fb in df.Loyalty_level:
   for word in fb.split():
      unique_words.add(word)
print(f" Unique words used ({len(unique_words)} total):")
print(sorted(unique_words))
```

o/p:



Union,Intersection and Difference by Country filtering "India" > Revenue 1000.

Code: Union = df[(df['Country']=='India')|(df['Revenue']>1000)]



```
Union = df[(df['Country']=='India')|(df['Revenue']>1000)]   # union Country = India | Revenue >1000
Union
```

	CustomerID	Age	Gender	Country	Avgorder_value	Total_orders	Last_purchase	Is_fraudlent	Preferred_category	Email_open_rate	...	Loyalty_type
6	CUST_5563	71	Female	India	14.04	11	47 days	0	Beauty	4.1	...	High Loyal
9	CUST_4192	27	Male	India	14.31	8	100 days	0	Sports	20.4	...	Moderate Loyal
13	CUST_3062	52	Other	India	15.09	15	297 days	0	Electronics	90.2	...	Moderate Loyal
15	CUST_2120	29	Female	India	15.31	7	104 days	0	Beauty	99.1	...	Low Loyal
16	CUST_1917	33	Male	India	15.34	6	301 days	0	Sports	39.0	...	Fraud
...	...	...	...	...	...	...	...	...	...	...	...	...
4990	CUST_9578	62	Male	Japan	93.19	16	332 days	0	Fashion	8.1	...	Low Loyal

Code: **Intersection = df[(df['Country']=='India') & (df['Revenue']>1000)]**

o/p:

```
Intersection = df[(df['Country']=='India') & (df['Revenue']>1000)]
Intersection[['Country','Revenue']]
```

	Country	Revenue
1050	India	1090.20
1218	India	1170.00
1338	India	1056.04
1770	India	1020.74
1777	India	1095.30
...	...	...
4907	India	1025.09
4942	India	1211.47
4955	India	1584.23
4958	India	1118.28
4977	India	1025.09

221 rows × 2 columns

Code: **Difference = df[(df['Country']=='India') &~ (df['Revenue'] > 1000)]**

o/p:

```
Difference = df[(df['Country']=='India') &~ (df['Revenue'] > 1000)]
Difference[['Country','Revenue']]
```

	Country	Revenue
6	India	154.44
9	India	114.48
13	India	226.35
15	India	107.17
16	India	92.04
...	...	...
4939	India	931.90
4973	India	838.71
4982	India	931.90
4986	India	745.52
4991	India	652.33

Groupby Revenue by Gender

Code: **revenue_gender = df.groupby('Gender')['Revenue'].sum()**

o/p:

```
revenue_gender = df.groupby('Gender')['Revenue'].sum()
revenue_gender
```

Gender	Revenue
Female	2292156.81
Male	2420816.62
Other	516681.85

dtype: float64

Groupby Loyalty_score by Is_fraudulent

Code: **df.groupby('Is_fraudulent')['Loyalty_score'].mean()**

o/p:

```
df.groupby('Is_fraudlent')['Loyalty_score'].mean()
```

Is_fraudlent	Loyalty_score
0	50.041900
1	50.713178

dtype: float64

Initial EDA for Cleaned dataset

1.Shape = Number of rows and columns.

Code: **df.shape**

o/p:

```
df.shape  # cleaned dataset

(4950, 24)
```

2.Columns = Name of each column.

Code: **df.columns**

o/p:

```
df.columns  # cleaned dataset column names

Index(['CustomerID', 'Age', 'Gender', 'Country', 'Avgorder_value',
       'Total_orders', 'Last_purchase', 'Is_fraudlent', 'Preferred_category',
       'Email_open_rate', 'Customer_since', 'Loyalty_score', 'churn_risk',
       'Revenue', 'Loyalty_type', 'Loyalty_bucket', 'Loyalty_or_fraud',
       'Is_weekend', 'Year', 'Risk_level', 'Loyalty_level', 'Country_Loyalty',
       'Customer_tag', 'Revenue_loyal'],
      dtype='object')
```

3.Numerical columns = Group of numerical datatype

Code: **df.select_dtypes(include=['int64','float64','bool','datetime64[ns]', 'timedelta64[ns]']).columns**

o/p:

```
df.select_dtypes(include=['int64','float64','bool','datetime64[ns]','timedelta64[ns]']).columns

Index(['Age', 'Avgorder_value', 'Total_orders', 'Last_purchase',
       'Is_fraudlent', 'Email_open_rate', 'Customer_since', 'Loyalty_score',
       'churn_risk', 'Revenue', 'Is_weekend'],
      dtype='object')
```

4.Categorical columns = Group of Category datatype

Code: **df.select_dtypes(include=['object','category']).columns**

o/p:

```
df.select_dtypes(include=['object','category']).columns    # categorical columns

Index(['CustomerID', 'Gender', 'Country', 'Preferred_category', 'Loyalty_type',
       'Loyalty_bucket', 'Risk_level', 'Loyalty_level', 'Country_Loyalty',
       'Customer_tag', 'Revenue_loyal'],
      dtype='object')
```

5.Information = Tells total number of columns, Null values, Data type

Code: **df.info()**

o/p:

```
 #   Column              Non-Null Count  Dtype
---  ------              --------------  -----
 0   CustomerID          4950 non-null   object
 1   Age                 4950 non-null   int64
 2   Gender              4950 non-null   object
 3   Country             4950 non-null   object
 4   Avgorder_value      4950 non-null   float64
 5   Total_orders        4950 non-null   int64
 6   Last_purchase       4950 non-null   timedelta64[ns]
 7   Is_fraudlent        4950 non-null   int64
 8   Preferred_category  4950 non-null   object
 9   Email_open_rate     4950 non-null   float64
 10  Customer_since      4950 non-null   datetime64[ns]
 11  Loyalty_score       4950 non-null   int64
 12  churn_risk          4950 non-null   float64
 13  Revenue             4950 non-null   float64
 14  Loyalty_type        4950 non-null   object
 15  Loyalty_bucket      4950 non-null   category
 16  Loyalty_or_fraud    4950 non-null   string
 17  Is_weekend          4950 non-null   bool
 18  Year                4950 non-null   int32
 19  Risk_level          4950 non-null   object
 20  Loyalty_level       4950 non-null   object
 21  Country_Loyalty     4950 non-null   object
 22  Customer_tag        4950 non-null   object
 23  Revenue_loyal       4950 non-null   object
dtypes: bool(1), category(1), datetime64[ns](1), float64(4), int32(1)
memory usage: 879.9+ KB
```

6.Data type = Shows data type of each column

Code: **df.dtypes**

o/p:

```
df.dtypes  # cleaned dataset datatypes of each column
```

	0
CustomerID	object
Age	int64
Gender	object
Country	object
Avgorder_value	float64
Total_orders	int64
Last_purchase	timedelta64[ns]
Is_fraudlent	int64
Preferred_category	object
Email_open_rate	float64
Customer_since	datetime64[ns]
Loyalty_score	int64
churn_risk	float64
Revenue	float64
Loyalty_type	object
Loyalty_bucket	category
Loyalty_or_fraud	string[python]
Is_weekend	bool
Year	int32
Risk_level	object
Loyalty_level	object
Country_Loyalty	object
Customer_tag	object
Revenue_loyal	object

dtype: object

7.Head = First five rows

Code: **df.head()**

o/p:

```
df.head()  # cleaned dataset first five rows
```

	CustomerID	Age	Gender	Country	Avgorder_value	Total_orders	Last_purchase	Is_fraudlent	Preferred_category	Email_open_rate	...	Loyalty_type	Loyal
0	CUST_9340	43	Female	Germany	10.66	13	346 days	0	Electronics	97.9	...	Fraud	
1	CUST_8604	44	Male	China	10.70	6	170 days	0	Sports	84.5	...	High Loyal	
2	CUST_2440	77	Male	Canada	11.48	9	245 days	0	Fashion	71.4	...	Moderate Loyal	
3	CUST_7530	78	Female	China	11.61	11	121 days	0	Sports	20.8	...	High Loyal	
4	CUST_2826	54	Male	China	12.50	10	182 days	0	Beauty	98.5	...	Low Loyal	

5 rows × 24 columns

8.Tail = Last five rows

Code: **df.tail()**

o/p:

```
df.tail()  # cleaned dataset last five rows
```

	CustomerID	Age	Gender	Country	Avgorder_value	Total_orders	Last_purchase	Is_fraudlent	Preferred_category	Email_open_rate	...	Loyalty_type
4995	CUST_9728	33	Female	Germany	93.19	12	278 days	0	Home	19.1	...	Fraud
4996	CUST_9801	76	Female	Canada	93.19	4	268 days	0	Electronics	81.1	...	Fraud
4997	CUST_9858	23	Female	Germany	93.19	8	193 days	0	Fashion	33.4	...	Moderate Loyal
4998	CUST_9970	21	Female	Canada	93.19	9	241 days	0	Electronics	97.3	...	High Loyal
4999	CUST_9984	68	Male	France	93.19	10	224 days	0	Beauty	43.2	...	High Loyal

5 rows × 24 columns

9. Statistical Column = Tells count, mean, minimum, maximum, std, 25%,50%,75% values of all numerical values.

Code: **df.describe().round(2)**

o/p:

```
df.describe().round(2)   # cleaned dataset summarize
```

	Age	Avgorder_value	Total_orders	Last_purchase	Is_fraudlent	Email_open_rate	Customer_since
count	4950.00	4950.00	4950.00	4950	4950.00	4950.00	4950
mean	48.14	104.79	10.03	180 days 00:49:27.272727272	0.03	50.68	2023-01-02 18:33:01.090908928
min	18.00	10.66	0.00	0 days 00:00:00	0.00	0.00	2020-06-29 00:00:00
25%	33.00	59.09	8.00	89 days 00:00:00	0.00	26.40	2021-09-19 06:00:00
50%	48.00	93.19	10.00	178 days 00:00:00	0.00	50.95	2023-01-12 00:00:00
75%	64.00	137.28	12.00	270 days 00:00:00	0.00	75.28	2024-04-12 00:00:00
max	79.00	336.67	23.00	364 days 00:00:00	1.00	100.00	2025-06-28 00:00:00
std	17.91	61.21	3.16	104 days 19:26:39.624362962	0.16	28.36	NaN

10.Sample = Any five rows display

Code: **df.sample(5)**

o/p:

```
df.sample(5)   # cleaned dataset any 5 rows by sample
```

	CustomerID	Age	Gender	Country	Avgorder_value	Total_orders	Last_purchase	Is_fraudlent	Preferred_category
2879	CUST_9966	71	Male	Canada	110.49	11	110 days	0	Electronics
2233	CUST_7136	70	Male	UK	87.81	8	221 days	0	Electronics
2383	CUST_1600	28	Female	Japan	93.58	6	27 days	0	Home
2249	CUST_7901	40	Male	USA	88.20	17	129 days	0	Home
4094	CUST_7663	61	Female	Brazil	180.71	10	189 days	0	Fashion

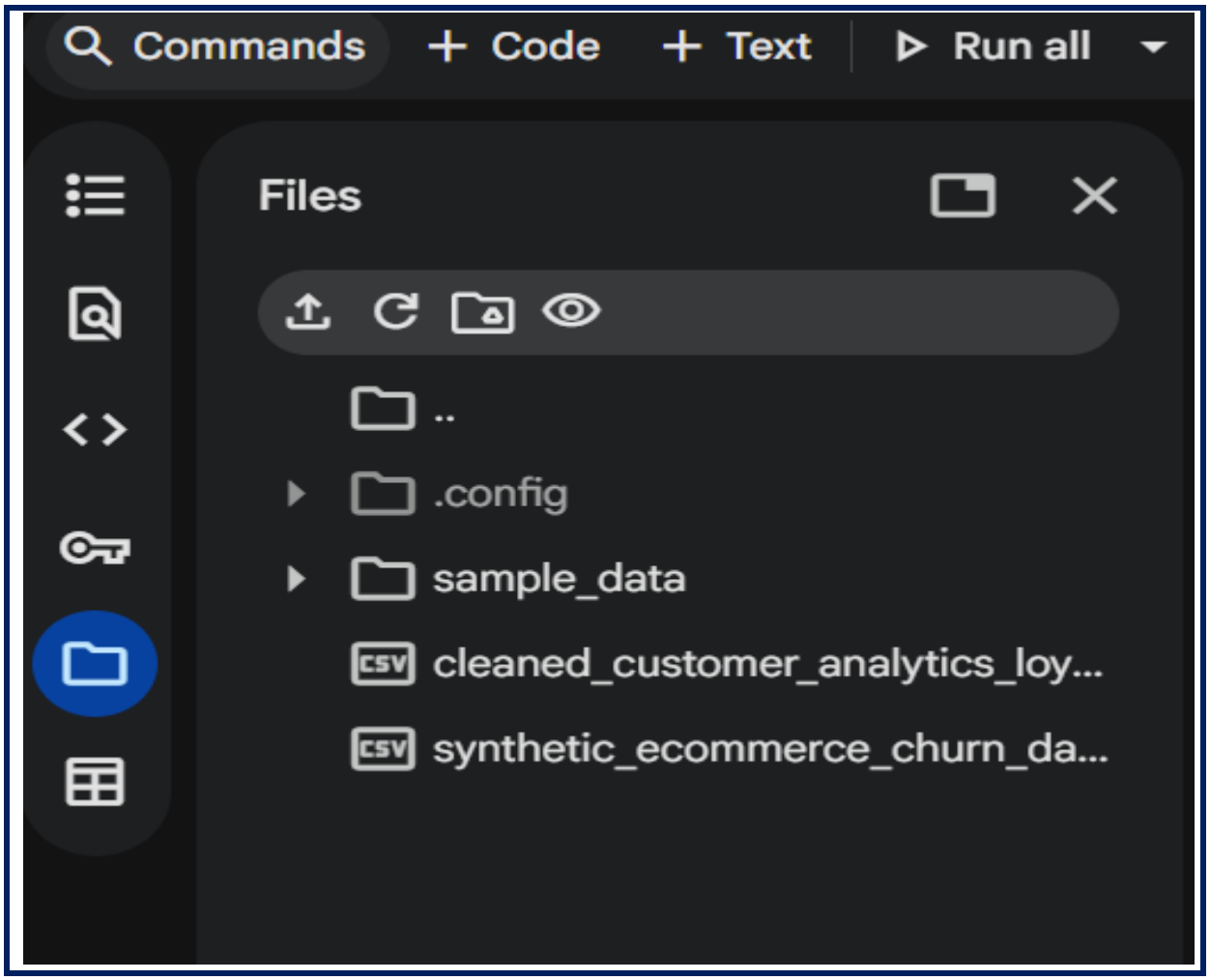
5 rows × 24 columns

Converting cleaned dataset to csv file:

Code:

df.to_csv('cleaned_customer_analytics_loyalty_vs_fraud.csv',index=False)

o/p:



Copy path of cleaned dataset is

/content/cleaned_customer_analytics_loyalty_vs_fraud.csv

Stage 3 – EDA and Visualizations

DASHBOARD -1

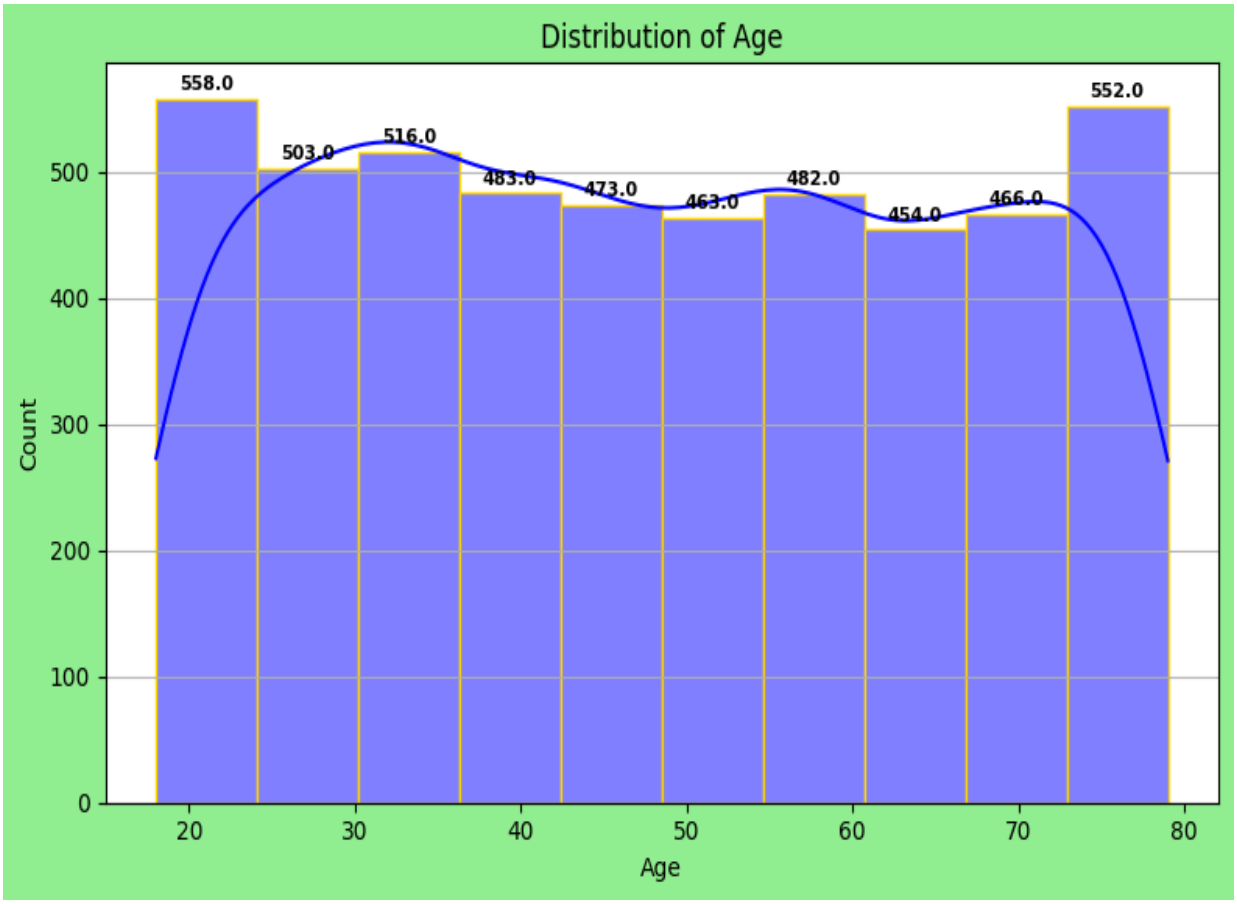
Univariate Analysis

1. Histogram (Distribution of Age)

Code:

```
# Univariate Analysis
plt.figure(figsize=(8,5),facecolor= 'lightgreen')
x=sns.histplot(data=df['Age'],bins=10,kde=True,color='blue',edgecolor='gold')
for container in x.containers:
  x.bar_label(container,fmt='%.1f',labeltype='edge',fontsize=8,fontweight='bold',
                                                              color='black',padding=2)
plt.title('Distribution of Age')
plt.xlabel('Age')
plt.grid(axis='y',linestyle='-',alpha=0.9)
plt.tight_layout()
plt.show()
```

o/p:



Interpretation of Histogram:

- **Title:**

  "Distribution of Age"

- **Explanation:**
  - Above chart is Univariate Analysis
  - This chart Histogram for Age distribution among customers.

- **Saying:**
  - Information about groups ages into bins and shows how many customers in each range.
  - Kde curve smooth distribution and shows age is almost uniform.

- **Features:**
  - X-axis = Age,
  - Y-axis = count of age.

- **Showing:**
  - The tallest bar indicates age range with highest customers and shows whether customer is young, middle or old age wise.
  - Bin edges showing age ranges are divided.
  - Blue line curve shows density of overall trend of age distribution.

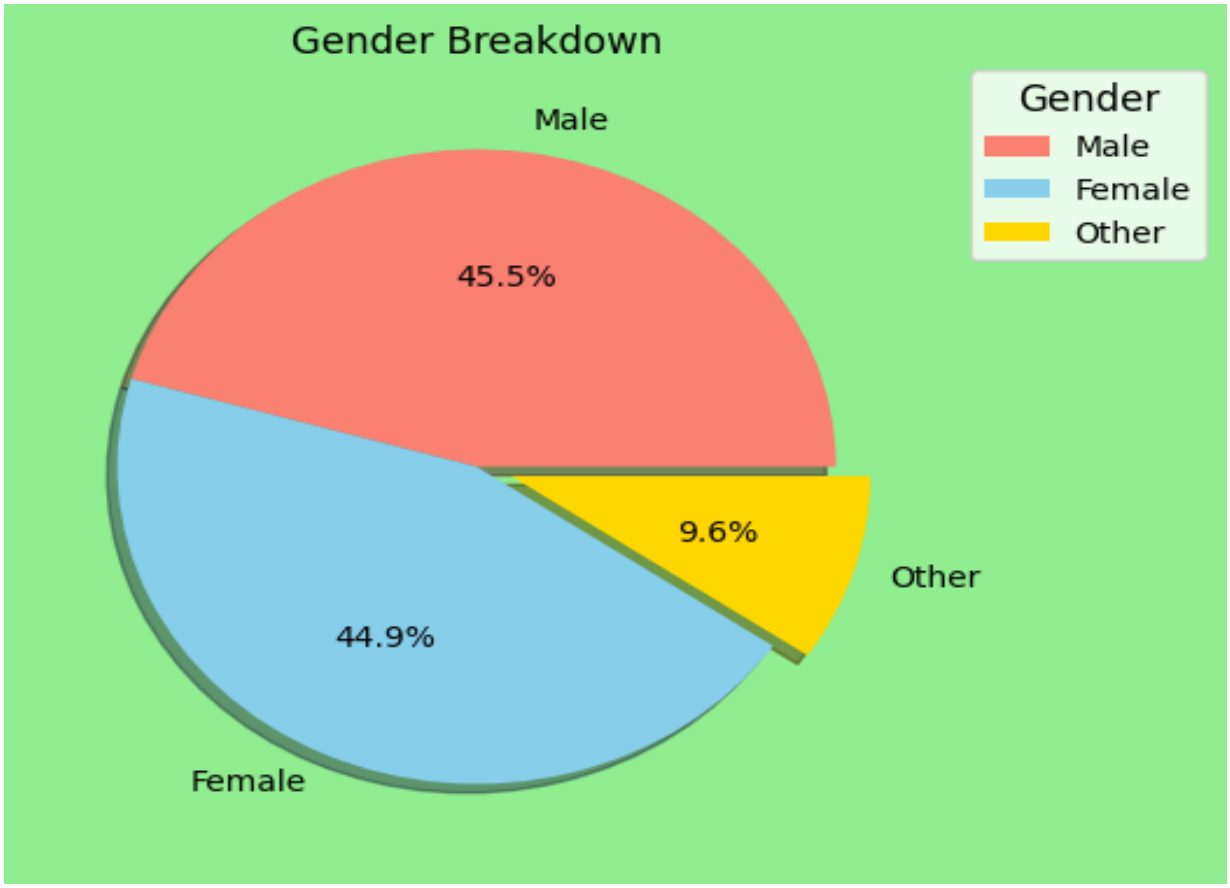
Univariate Analysis

2. Pie chart (Gender Breakdown)

Code:

```
# Univariate Analysis
gender_counts = df['Gender'].value_counts()
plt.figure(figsize=(8,5),facecolor= 'lightgreen')
plt.pie(gender_counts.values,
      labels=gender_counts.index,
      colors=['salmon','skyblue','gold'],
      autopct="%1.1f%%",
       explode=[0,0,0.1],
      shadow=True
      )
plt.legend(title='Gender',title_fontsize=12,fontsize=10,bbox_to_anchor=(1.05, 1),
                                                        loc='upper left',borderaxespad=0)
plt.title('Gender Breakdown')
plt.show()
```

o/p:



Interpretation of Pie chart:

- **Title:**

  "Gender Breakdown"

- **Explanation:**
  - Above chart is Univariate Analysis.
  - This above Pie chart is for Gender count of the customers.

- **Saying:**
  - Pie chart of gender breakdown saying that male customers are slightly high when compare female customers.
  - Here others represent lower number of values may be kids or transgenders.

- **Features:**
  - Gender column.

- **Showing:**
  - Clear percentage.6 values of male, female and other customers.

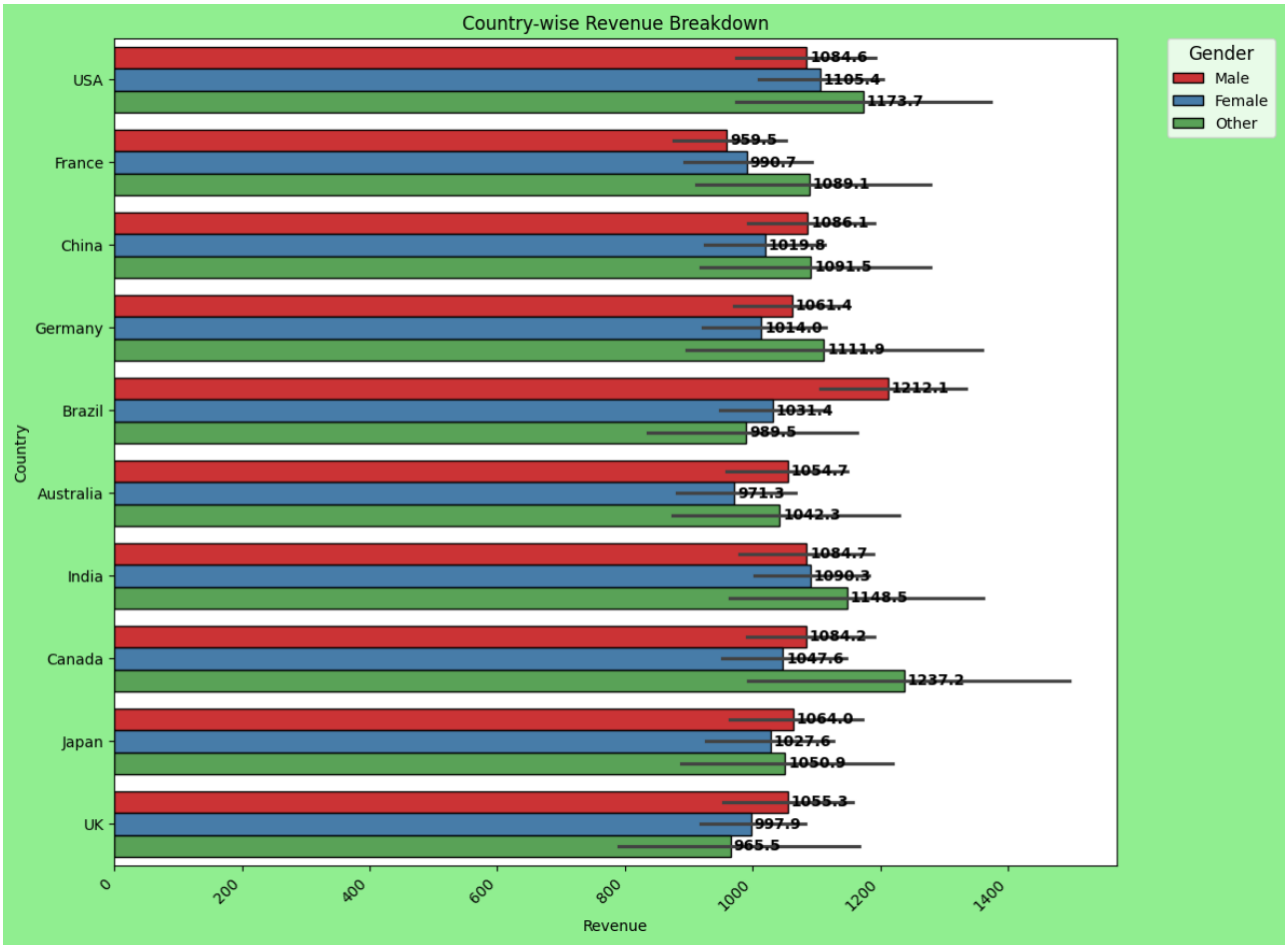
Bivariate Analysis

3. Bar chart (Country-wise Revenue Breakdown)

Code:

```
# Bivariate Analysis
df_sorted = df.sort_values('Revenue',ascending=False)
plt.figure(figsize=(12,10),facecolor='lightgreen')
x=sns.barplot(data=df_sorted,x='Revenue',y='Country',hue= 'Gender',palette='Set1',
                                                    edgecolor='black',alpha=1.0)
plt.title('Country-wise Revenue Breakdown')
plt.xlabel('Revenue')
plt.xticks(rotation=45,ha='right')
plt.ylabel('Country')
for container in x.containers:
  x.bar_label(container,fmt='%.1f',label_type='edge',fontsize=10,
                                        fontweight='bold', color='black',padding=2)
plt.legend(title='Gender',title_fontsize=12,fontsize=10,bbox_to_anchor=(1.05, 1),
                                                    loc='upper left',borderaxespad=0)
plt.show()
```

o/p:



Interpretation of Bar chart

- **Title:**

  "Country-wise Revenue Breakdown"

- **Explanation:**
  - Above chart is Bivariate Analysis.
  - This above bar chart explains about Revenue among country-wise including gender categories.

- **Saying:**
  - Information about how much amount of revenue collected by each country based on their genders.
  - Revenue details for each country are mentioned with gender wise male, female and other revenue separately.

- **Features:**
  - X-axis = Revenue,
  - Y-axis = Country,
  - hue = Gender.

- **Showing:**
  - Bars are clearly plotted to each country according to their revenue by gender-wise.
  - Canada Female customers revenue is high and also Brazil Male customers revenue value is high when compare to other country genders.
  - UK and France customers having low revenue when compare to other country peoples.

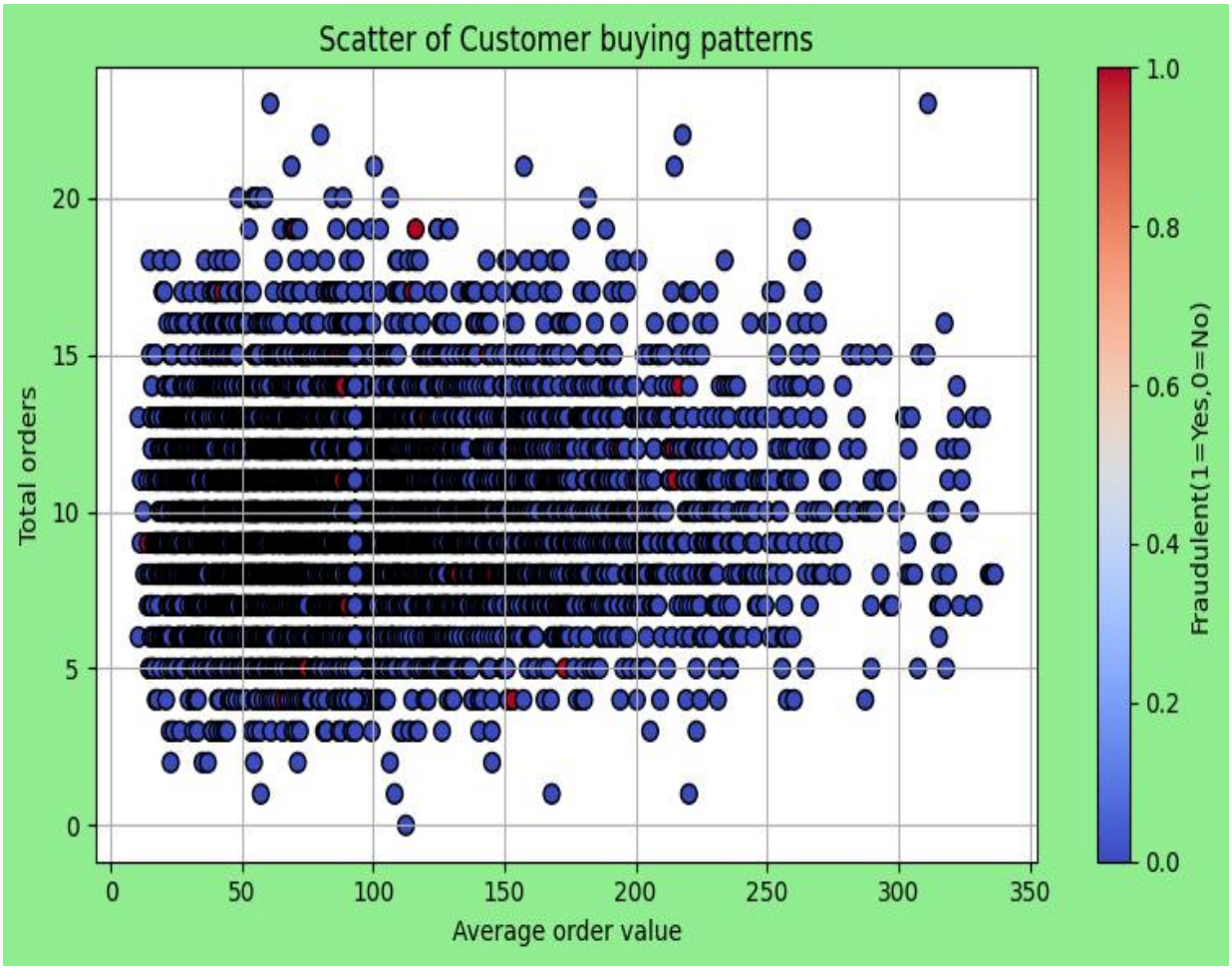
Bivariate Analysis.

4. Scatter plot (Scatter of Customer buying patterns)

Code:

```
# Bivariate Analysis.
plt.figure(figsize=(8,5),facecolor='lightgreen')
plt.scatter(df.Avgorder_value,df.Total_orders,s=50,c=df.Is_fraudulent,cmap='coolwarm',
                                                    edgecolor= 'black',marker='o',alpha=1.0)
plt.title('Scatter of Customer buying patterns')
plt.xlabel('Average order value')
plt.ylabel('Total orders')
plt.grid(True)
plt.colorbar(label='Fraudulent(1=Yes,0=No)')
plt.tight_layout()
plt.show()
```

o/p:



Interpretation of Scatter Chart

- **Title:**

"Scatter of Customer buying patterns"

- **Explanation:**
  - Above chart is Bivariate Analysis.
  - Above Scatter chart explains about the value spreads between average order value and total orders.

- **Saying:**
  - Information of customers among total order and average order value whether is fraudulent or not.
  - Dot represents customers.

- **Features:**
  - X-axis = Avg_order_value,
  - Y-axis = Total orders.

- **Showing:**
  - Shows an analysis about customer buying patterns.
  - Most points are clustered in the lower and mid ranges.

Multivariate Analysis

5. Heatmap (Customer Activity Matrix)

Code:

```
ncol=df.select_dtypes('int').columns
corr_matrix = df[ncol].corr()
corr_matrix
```

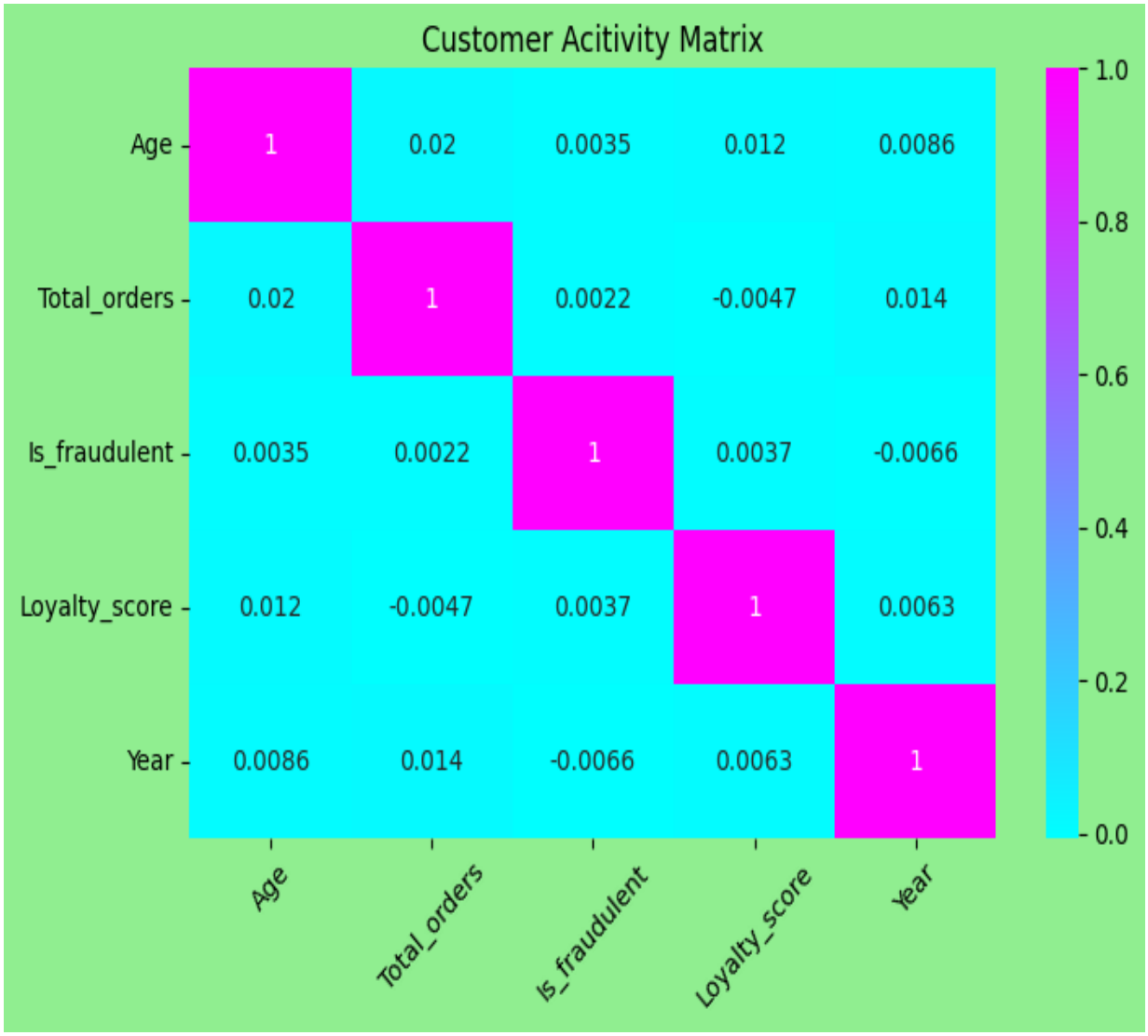
o/p:

	Age	Total_orders	Is_fraudulent	Loyalty_score	Year
Age	**1.000000**	**0.020457**	**0.003486**	**0.011693**	**0.008580**
Total_orders	**0.020457**	**1.000000**	**0.002187**	**-0.004702**	**0.013656**
Is_fraudulent	**0.003486**	**0.002187**	**1.000000**	**0.003707**	**-0.006612**
Loyalty_score	**0.011693**	**-0.004702**	**0.003707**	**1.000000**	**0.006341**
Year	**0.008580**	**0.013656**	**-0.006612**	**0.006341**	**1.000000**

Code:

```
#Multivariate Analysis
plt.figure(figsize=(8,5),facecolor='lightgreen')
sns.heatmap(corr_matrix,cmap='cool',annot=True)
plt.title('Customer Acitivity Matrix')
plt.xticks(rotation=45)
plt.show()
```

o/p:


Customer Acitivity Matrix
Age
Total_orders
Is_fraudulent
Loyalty_score
Year
1
0.02
0.0035
0.012
0.0086
0.02
1
0.0022
-0.0047
0.014
0.0035
0.0022
1
0.0037
-0.0066
0.012
-0.0047
0.0037
1
0.0063
0.0086
0.014
-0.0066
0.0063
1
1.0
0.8
0.6
0.4
0.2
0.0

Interpretation of Heatmap

- **Title:**
  "Customer Activity Matrix"

- **Explanation:**
  - Above chart is Multivariate Analysis.
  - Heatmap is the correlation between different columns and number inside tells strongly how two variables are connected.

- **Saying:**
  - Is_fraudulent has almost no correlation.
  - Age, Total orders, Loyalty score is almost un related.
  - Year and Month have a negative relationship.
  - Diagonals are 1.

- **Features:**
  - Age, Total orders, Loyalty score, Year, Month, Is_fraudulent.

- **Showing:**
  - Shows all diagonal values are 1 which means every variable is perfectly correlated with itself.

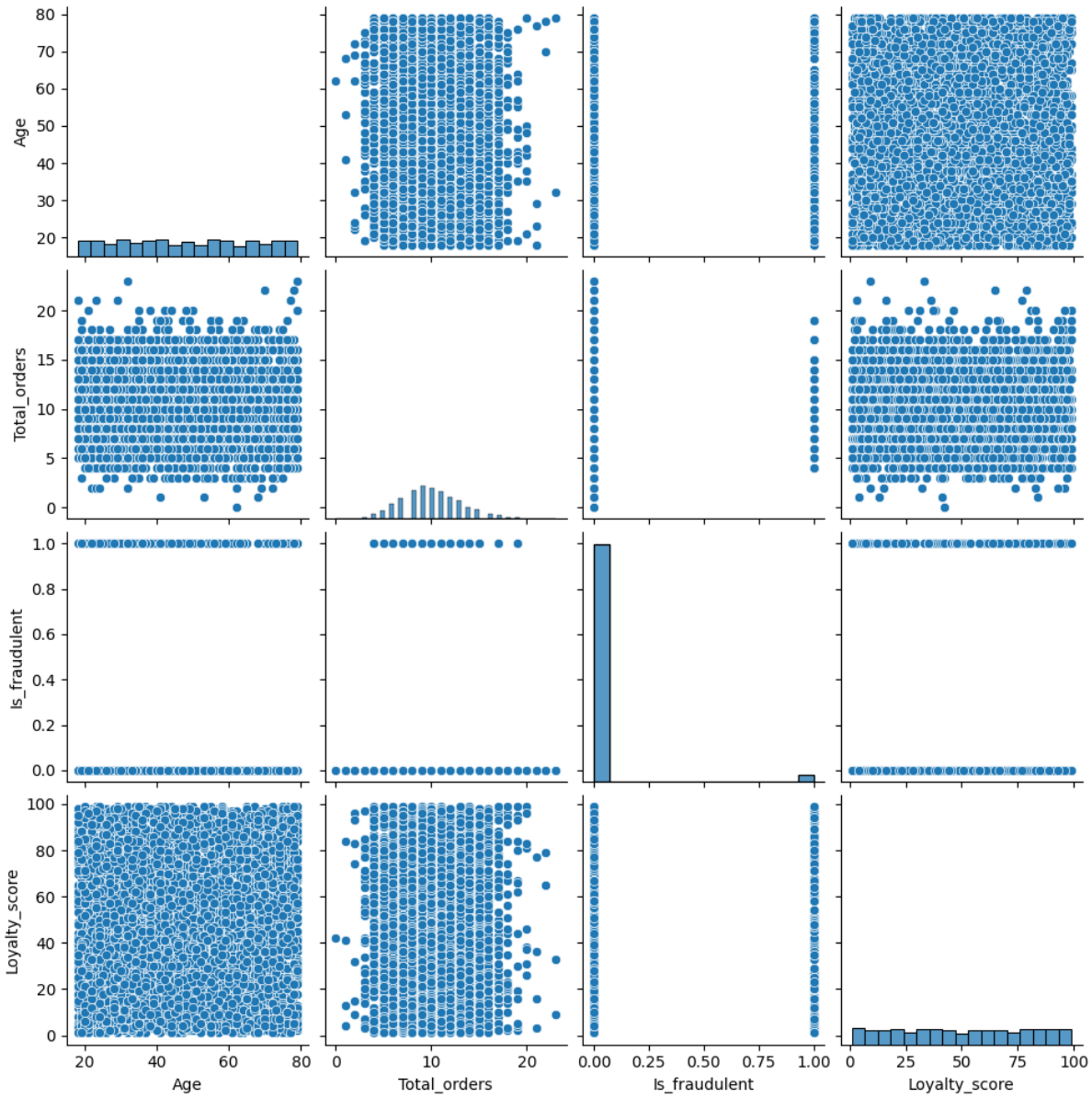
Multivariate Analysis

6. Pair Plot

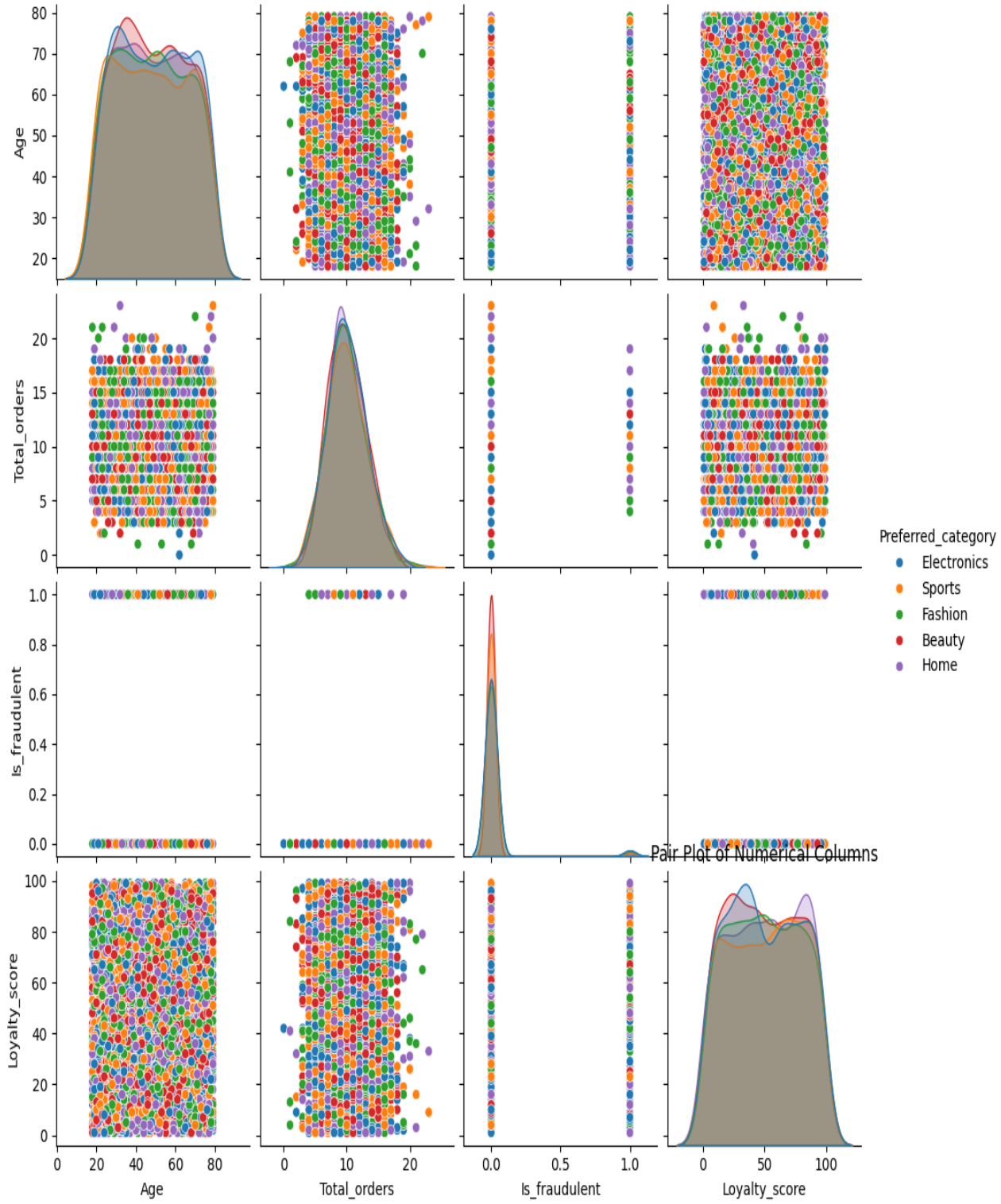
Code:

```
# Multivariate Analysis
ncol=df.select_dtypes('n').columns
sns.pairplot(df[ncol])
sns.pairplot(data=df,vars=ncol,hue='Preferred_category')
plt.title('Pair Plot of Numerical Columns')
plt.show()
```

o/p:



Customer Analytics Loyalty Vs Fraud



Interpretation of Pair Plot

- **Title:**

  "Pair Plot of Numerical Columns."

- **Explanation:**
  - Above chart is Multivariate Analysis.
  - Pair plot helps to identify patterns, distributions and correlations across multiple variables in one combined view.
- **Saying:**
  - It saying how different numerical features relate to each other across customer preference categories.
- **Features:**
  - All Numerical columns.
- **Showing:**
  - Diagonal plots show distribution of individual numerical features.
  - Off-diagonal scatter plots show relationships between features such as Age, Total Orders, Revenue.
  - Dense clusters show common behavioural patterns among customers.
  - Spread-out areas reflect variability across categories.

DASHBOARD-2

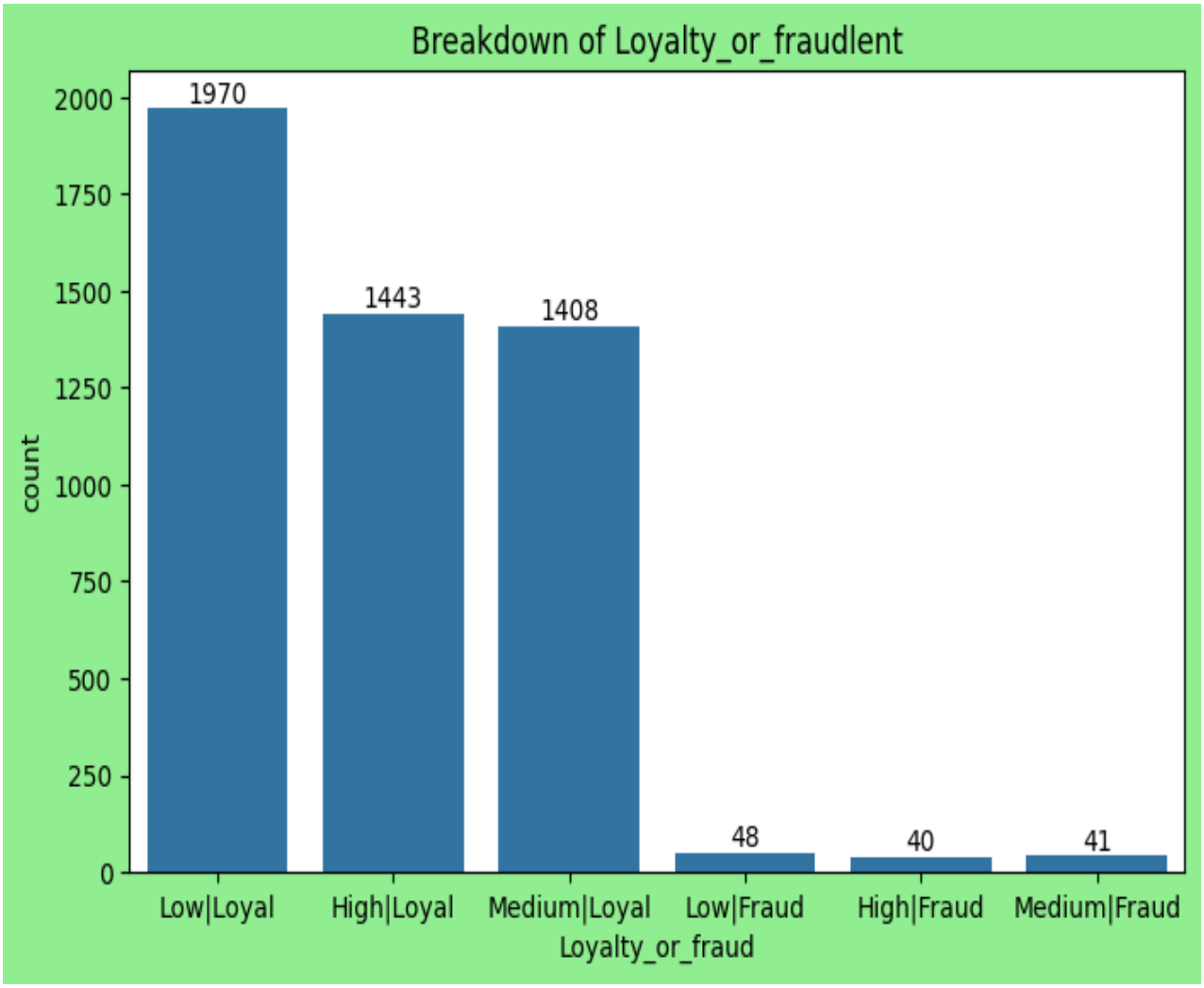
Univariate Analysis

7.Count plot (Breakdown of Loyalty or fraudulent)

Code:

```
# Univariate Analysis
plt.figure(figsize=(8,5),facecolor='lightgreen')
cp=sns.countplot(data=df,x='Loyalty_or_fraud')
cp.bar_label(cp.containers[0])
plt.title('Breakdown of Loyalty_or_fraudlent')
plt.show()
```

o/p:



Interpretation of Count plot

- **Title:**
  "Breakdown of Loyalty_or_fraudlent"
- **Explanation:**
  - Above chart is Univariate Analysis
  - Count plot is for Loyalty or fraud calculation.
- **Saying:**
  - Information about how many customers is loyal or fraud based on low, medium or high.
  - Very low numbers of fraud customers even two digits only are present in all low fraud, medium fraud and High fraud.
  - Loyal customer in both medium loyal and high loyal are almost similar, when compare to low loyal customers,
- **Features:**
  - X-axis = Loyalty_or_fraud,
  - Y-axis = count of Loyalty_or_fraud.
- **Showing:**
  - Count-wise clear explanation about customers is loyal or fraud.

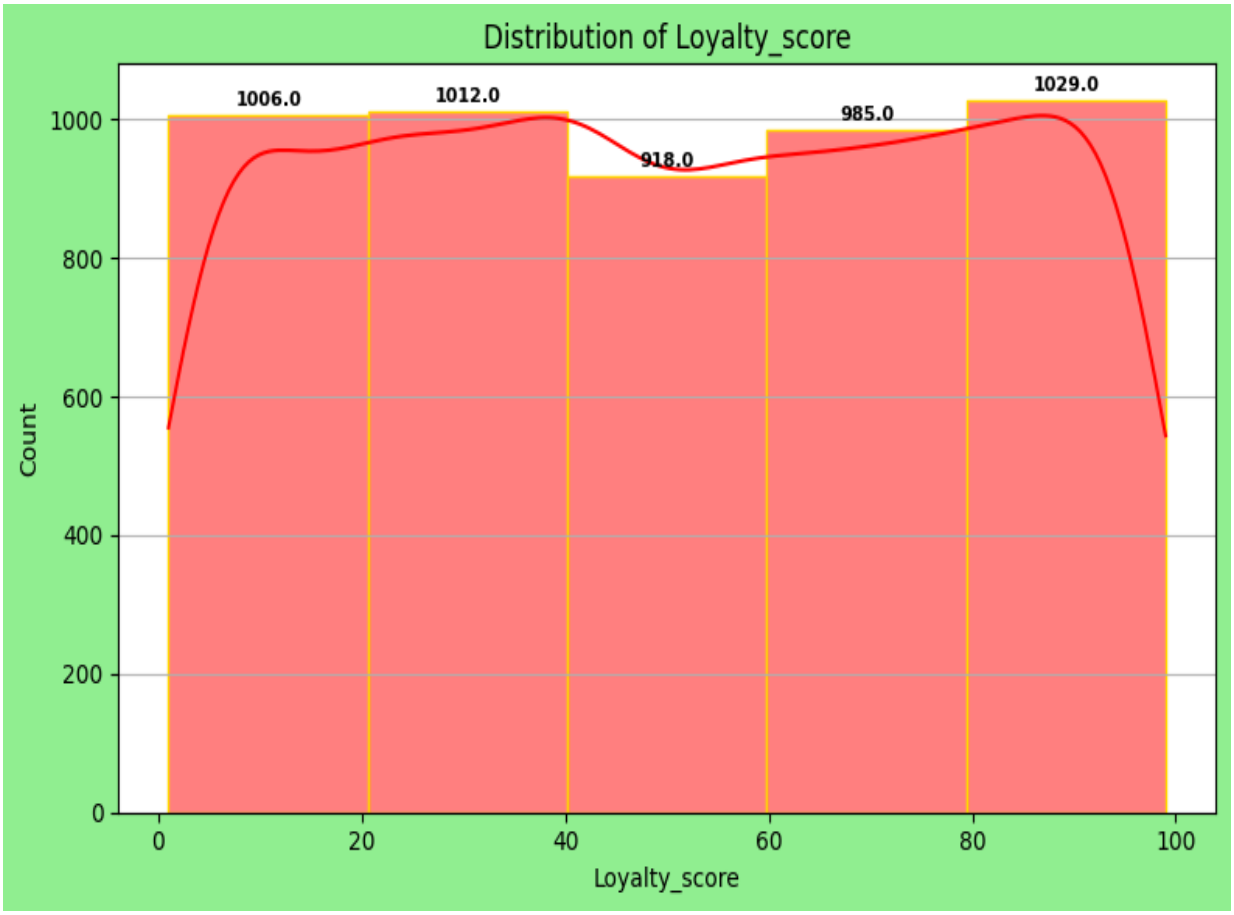
Univariate Analysis

8. Histogram (Distribution of Loyalty score)

Code:

```
# Univariate Analysis
plt.figure(figsize=(8,5),facecolor='lightgreen')
x=sns.histplot(data=df['Loyalty_score'],bins=5,kde= True,edgecolor='gold',color='red')
for container in x.containers:
  x.bar_label(container,fmt='%.1f',label_type='edge',fontsize=8,fontweight='bold',
                                                              color='black',padding=2)
plt.title('Distribution of Loyalty_score')
plt.xlabel('Loyalty_score')
plt.grid(axis='y',linestyle='-',alpha=0.9)
plt.tight_layout()
plt.show()
```

o/p:



Interpretation of Histogram chart

- **Title:**
  "Distribution of Loyalty_score"

- **Explanation:**
  - Above chart is Univariate Analysis
  - Based on the loyalty_score of each customer, histogram chart is done.

- **Saying:**
  - Above images tells how loyalty score spreads among customers.
  - Kde smooth curve that shows the overall shape of the score distribution.

- **Features:**
  - X-axis = Loyalty_score,
  - Y-axis = Count of Loyalty_score.

- **Showing:**
  - Distribution among customers who are loyal based on loyalty score spread evenly but slight peak around 10,40 and 90.
  - Red line shows the distribution of score.
  - The curve shows scores are most concentrated.

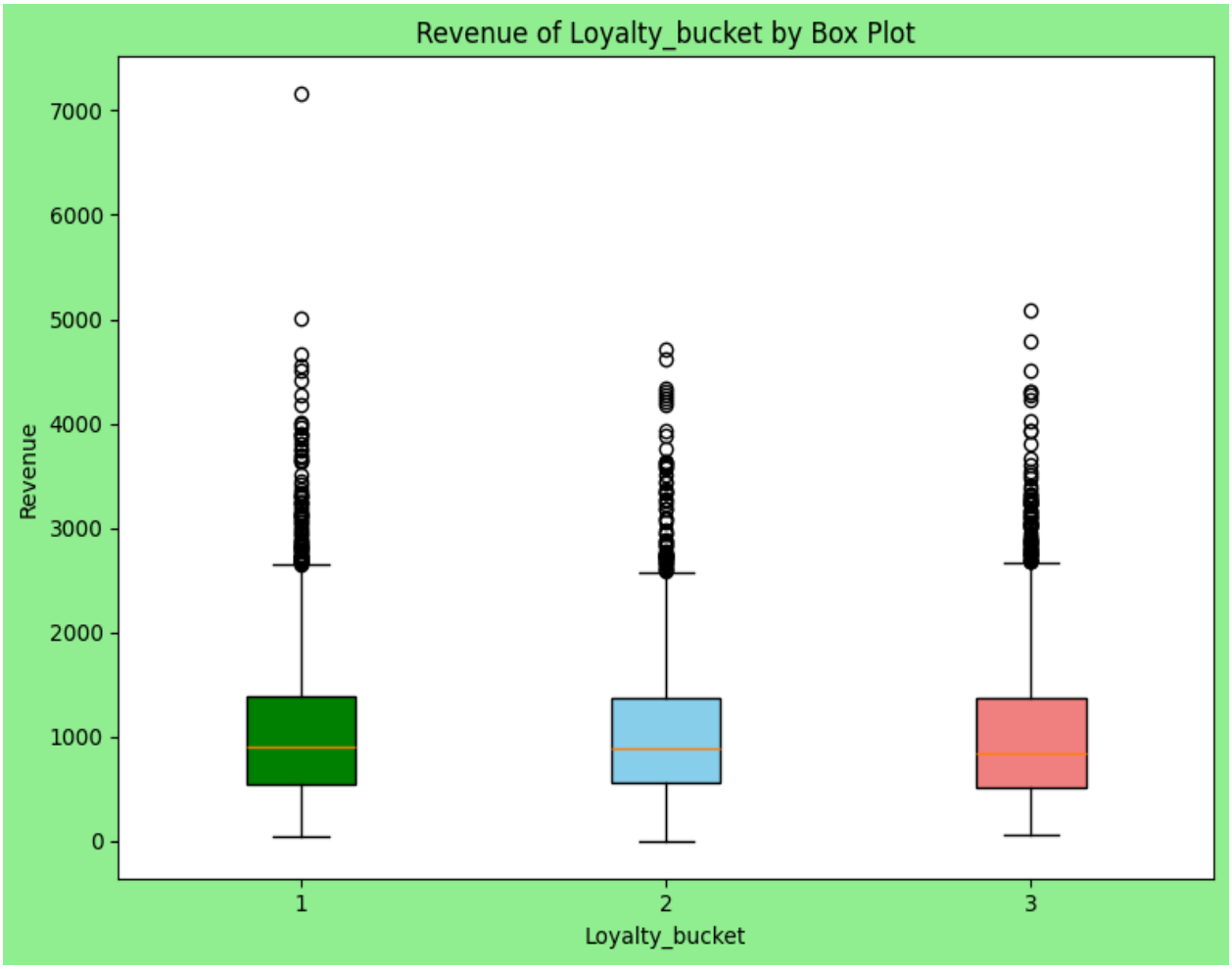
Bivariate Analysis

9. Box plot (Loyalty_bucket by Revenue)

Code:

```
# Bivariate Analysis
plt.figure(figsize=(8,6),facecolor='lightgreen')
low=df[df['Loyalty_bucket']=='Low']['Revenue']
medium=df[df['Loyalty_bucket']=='Medium']['Revenue']
high=df[df['Loyalty_bucket']=='High']['Revenue']
colors=['green','skyblue','lightcoral']
bp=plt.boxplot([low,medium,high],patch_artist=1)
for patch,color in zip(bp['boxes'],colors):
  patch.set_facecolor(color)
plt.title('Revenue of Loyalty_bucket by Box Plot')
plt.xlabel('Loyalty_bucket')
plt.ylabel('Revenue')
plt.tight_layout()
plt.show()
```

o/p:



Interpretation of Box Plot

- **Title:**
  "Revenue of Loyalty bucket by Box plot"

- **Explanation:**
  - Above chart is Bivariate Analysis.
  - This plot compares Revenue across three Loyalty buckets Low, Medium and High Loyalty.

- **Saying:**
  - When revenue increases then loyalty also increases, strong +ve relationship between them.

- **Features:**
  - X-axis = Loyalty_bucket,
  - Y-axis = Revenue.
  -

- **Showing:**
  - Shows middle line inside box is Median revenue,25th and 75th percentile is inside box.
  - Minimum and Maximum ranges are whiskers.
  - Outliers are above the whiskers.

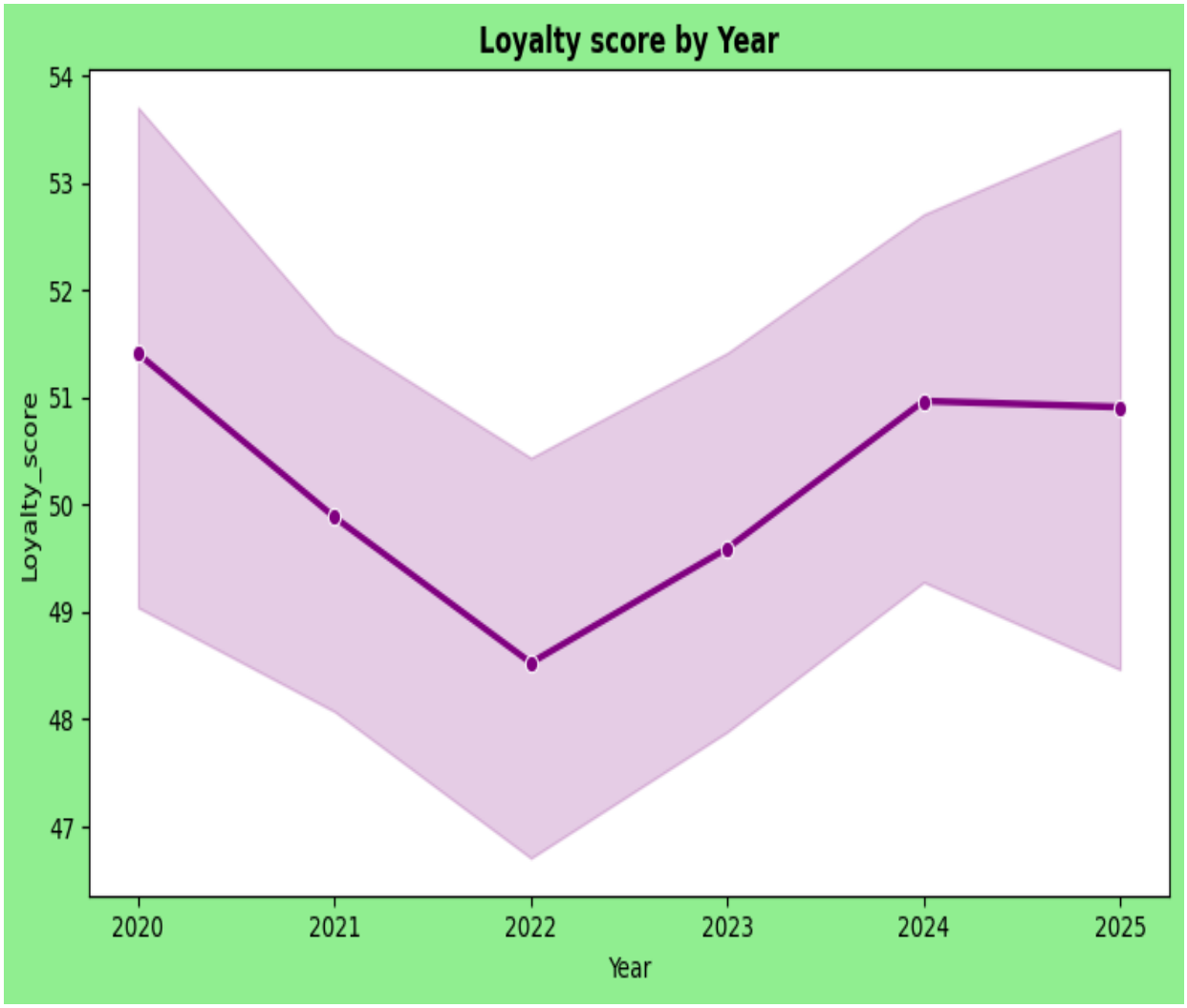
Bivariate Analysis.

10. Line plot (Loyalty score by Year)

Code:

```
# Bivariate Analysis.
plt.figure(figsize=(8,5),facecolor='lightgreen')
sns.lineplot(data=df,x='Year',y='Loyalty_score',color='purple',marker='o',linewidth=2.5)
plt.title('Loyalty score by Year',weight='bold',size=12)
plt.xlabel('Year')
plt.ylabel('Loyalty_score')
plt.tight_layout()
plt.show()
```

o/p:



Interpretation of Line Chart

- **Title:**
  "Loyalty score by Year"

- **Explanation:**
  - Above chart is Bivariate Analysis.
  - Line chart explains about Trend from 2020 to 2025 about average loyalty score.

- **Saying:**
  - Loyalty score is dropped from 2020 to 2022, then slighty improves at 2023 onwards
  - Finally stabilize at 2024 &2025

- **Features:**
  - X-axis = Year,
  - Y-axis = Loyalty_score.

- **Showing:**
  - 2020-2022 = Line is decline,
  - 2023 onwards = Improves,
  - 2024 & 2025 = Recovered and stabilized.

Bivariate Analysis

11. Bar Chart (Weekend Loyalty Score)

Code:

```
# Bivariate Analysis
plt.figure(figsize=(8,5),facecolor='lightgreen')
x=sns.barplot(data=df,x='Is_weekend',y='Loyalty_score',hue= 'Loyalty_bucket',
                                                    palette='Set1',edgecolor='gold',alpha=1.0)
for container in x.containers:
  x.bar_label(container, fmt='%.3f', label_type='edge', fontsize=9,
                                                    fontweight='bold',  color='black', padding=2)
plt.title('Weekend Loyalty score')
plt.xlabel('Is_weekend')
plt.xticks(rotation=60)
plt.ylabel('Loyalty_score')
plt.legend(title='Loyalty_bucket',title_fontsize=12,fontsize=10,
                                    bbox_to_anchor=(1.05, 1), loc='upper left',borderaxespad=0)
plt.show()
```

o/p:

Interpretation of Bar Chart

- **Title:**
  "Weekend Loyalty score"

- **Explanation:**
  - Bar plot is used for loyal customer in weekend days based on loyalty bucket are Low, Medium, High customers.

- **Saying:**
  - Clear view about the customers who are loyal at weekend days.

- **Features:**
  - X-axis = Is_weekend
  - Y-axis = Loyalty_score,
  - hue = Loyalty_bucket.

- **Showing:**
  - High customers are high in weekends,
  - Medium customers are next level,
  - Low customers are average but fraudulent are very less at weekends.

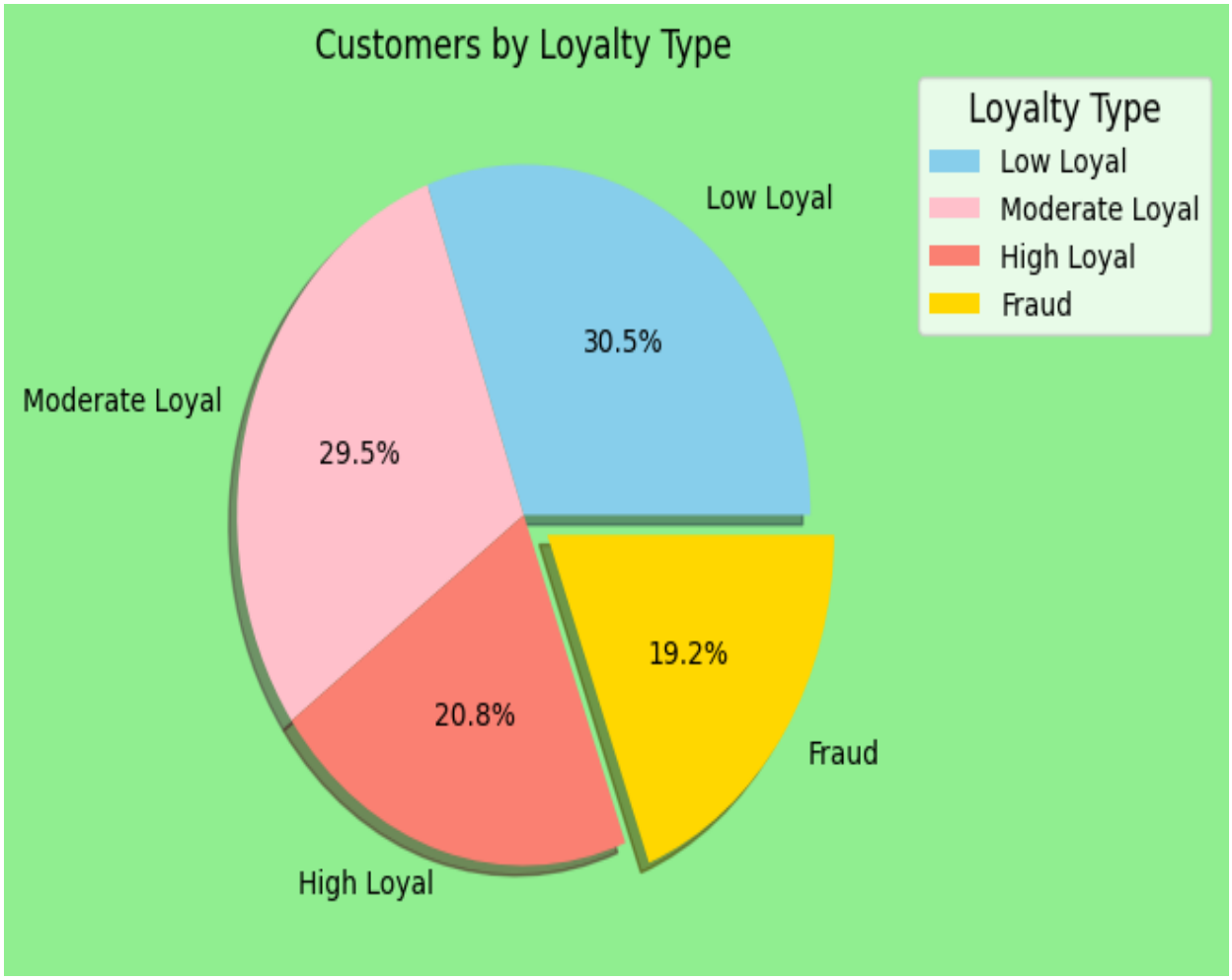
Univariate Analysis

12. Pie chart (Customers by Loyalty Type)

Code:

```
# Univariate Analysis
pie_loyalty=df['Loyalty_type'].value_counts()
plt.figure(figsize=(8,5),facecolor='lightgreen')
plt.pie(pie_loyalty,
        labels=pie_loyalty.index,
        colors=['skyblue','pink','salmon','gold'],
        autopct="%1.1f%%",
         explode=[0,0,0,0.1],
        shadow=True
        )
plt.title('Customers by Loyalty Type')
plt.legend(title='Loyalty      Type',title_fontsize=12,fontsize=10,bbox_to_anchor=(1.05,      1),
loc='upper left',borderaxespad=0)
plt.show()
```

o/p:



Interpretation of Pie Chart

- **Title:**
  "Customers by Loyalty Type"

- **Explanation:**
  - Above chart is Univariate Analysis.
  - Pie chart created based on Loyalty type.

- **Saying:**
  - Low loyal have high value, Moderate loyal is slightly less,
  - High loyal customers are next, fraud customers are less only.

- **Features:**
  - Loyalty_type column.

- **Showing:**
  - Shows visual pie charts with clear values.

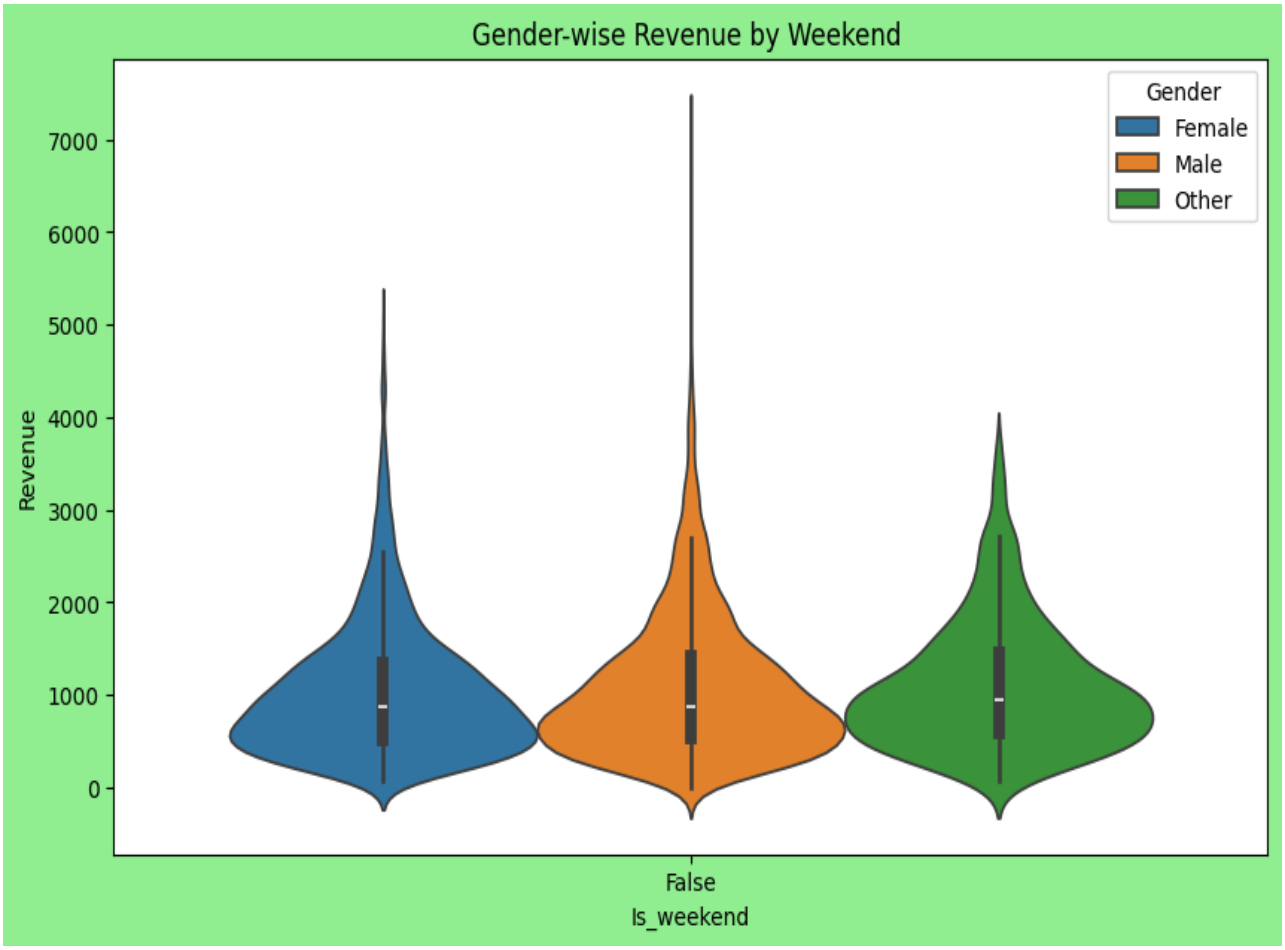
Bivariate Analysis

13. Violin Plot (Gender-wise Revenue by Weekend)

Code:

```
# Bivariate Analysis
plt.figure(figsize=(10,6),facecolor='lightgreen')
sns.violinplot(x='Is_weekend',y='Revenue',hue='Gender',data=df)
plt.title('Gender-wise Revenue by Weekend ')
plt.show()
```

o/p:



Interpretation of Violin Plot

- **Title:**
  "Gender-wise Revenue by Weekend."

- **Explanation:**
  - Above chart is Bivariate Analysis.
  - Violin plot combines box plot and density plot together.
  - The box inside the violin shows middle line = median, box area = IQR (Interquartile Range), Vertile line = overall range.

- **Saying:**
  - It tells the data is spread, where is dense, where it is less.
  - The wider part of the violin = more customers with revenue.
  - The narrower part = fewer customers with revenue.

- **Features:**
  - X-axis = Is_weekend,
  - Y-axis = Revenue,
  - hue = Gender.

- **Showing:**
  - First violin has a sharp thin tail, customers have very high revenue and has the lowest spread, meaning their weekend spending is more consistent.
  - Second violin is wider shape and median is slightly high and has a long upper tail.
  - Third violin has moderate spread and shows mid-level spread.

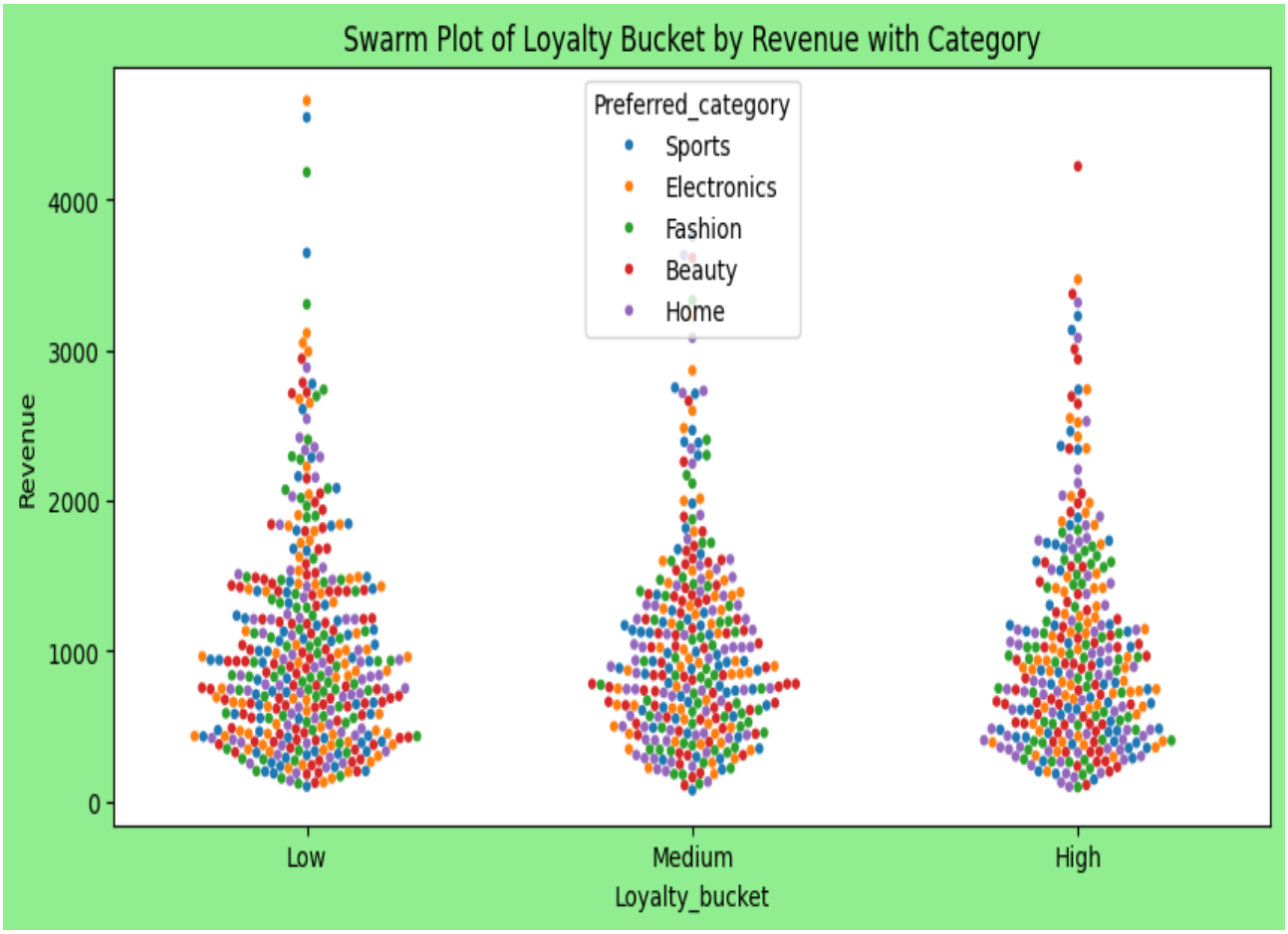
Bivariate Analysis

14. Swarm Plot (Loyalty Bucket by Revenue with Category)

Code:

```
# Bivariate Analysis
subset_df=df.sample(1000)
plt.figure(figsize=(10,5),facecolor='lightgreen')
sns.swarmplot(data=subset_df,x='Loyalty_bucket',y='Revenue',hue='Preferred_category',
                                                                                size=4)
plt.title('Swarm Plot of Loyalty Bucket by Revenue with Category')
plt.show()
```

o/p:



Interpretation of Swarm Plot

- **Title:**

  "Loyalty Bucket by Revenue with Category."

- **Explanation:**
  - Above chart is Bivariate Analysis.
  - Swarm plot spreads the points so none overlap, showing true distribution of revenue for each loyalty bucket level.

- **Saying:**
  - Customers with higher loyalty levels generally cluster around higher revenue.
  - Lower loyalty levels have many customers with lower and mid-level revenue.
  - At high loyalty levels, narrow but taller distribution means few customers but high in spreading.

- **Features:**
  - X-axis = Loyalty_bucket,
  - Y-axis = Revenue,
  - hue = Preferred_category.

- **Showing:**
  - Height of the dots represents the amount of revenue generated.
  - Taller cluster = more spread in revenue.
  - Wider cluster = more customer in loyalty bucket.

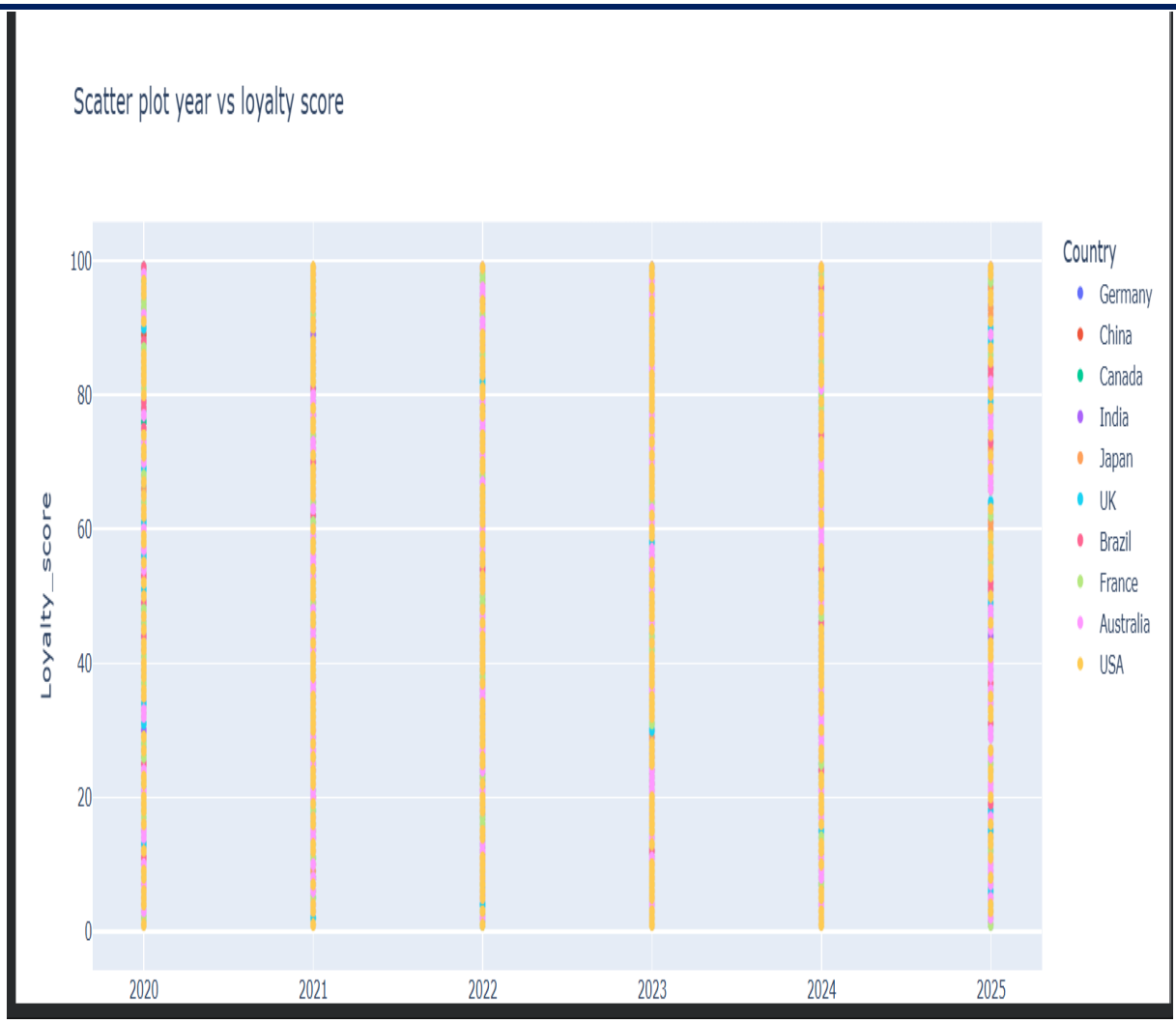
Bivariate Analysis with Interactive charts

15. Scatter Plot (Interactive chart)

Code:

```
#Bivariate Analysis with Interactive charts
import plotly.express as px
figure = px.scatter(df,x='Year',y='Loyalty_score',color='Country',
                                                    title='Scatter plot year vs loyalty score')
figure.show()
```

o/p:



Interpretation of Scatter Plot (Interactive chart)

- **Title:**

  "Scatter plot year vs loyalty score."

- **Explanation:**
  - Above chart is Bivariate Analysis with Interactive charts
  - Scatter plot interactive chart explains about loyalty score across years for different countries.

- **Saying:**
  - Different countries have different loyalty behaviours and there is no strong clear pattern like increasing or decreasing trend.

- **Features:**
  - X-axis = Year,
  - Y-axis = Loyalty_score,
  - hue = country.

- **Showing:**
  - Country colours help differentiate which country each dot belongs to.
  - Dots scattered from y-axis around 30 to 100 representing loyalty score.

Multivariate Analysis

16.Grouped Analysis

Code:

```
# Multivariate Analysis
group_summary=df.groupby(['Loyalty_bucket','Loyalty_type'])['Revenue'].agg
                                                    (['mean', 'median', 'sum', 'count']).reset_index()
group_summary
```

o/p:

	Loyalty_bucket	Loyalty_type	mean	median	sum	count
0	Low	Fraud	1059.653319	891.900	1008789.96	952
1	Low	High Loyal	NaN	NaN	0.00	0
2	Low	Low Loyal	1064.880197	913.240	1135162.29	1066
3	Low	Moderate Loyal	NaN	NaN	0.00	0
4	Medium	Fraud	NaN	NaN	0.00	0
5	Medium	High Loyal	NaN	NaN	0.00	0
6	Medium	Low Loyal	1008.391538	862.155	445709.06	442
7	Medium	Moderate Loyal	1080.663893	898.920	1088228.54	1007
8	High	Fraud	NaN	NaN	0.00	0
9	High	High Loyal	1068.546074	866.250	1099533.91	1029
10	High	Low Loyal	NaN	NaN	0.00	0
11	High	Moderate Loyal	996.104670	816.870	452231.52	454

Interpretation of Grouped Analysis

- **Title:**
  "Loyalty Bucket & Type vs Revenue"

- **Explanation:**
  - Loyalty type creates variation inside each bucket.
  - High bucket with premium type is the best revenue group.
  - Low bucket with basic type is the worst performing group.

- **Saying:**
  - High loyalty customers with premium loyalty type gives high revenue compared to all other groups.

- **Features:**
  - Loyalty_bucket,
  - Loyalty_type,
  - Revenue.

- **Showing:**
  - Revenue consistency increases as loyalty bucket level goes up.

DASHBOARD-3

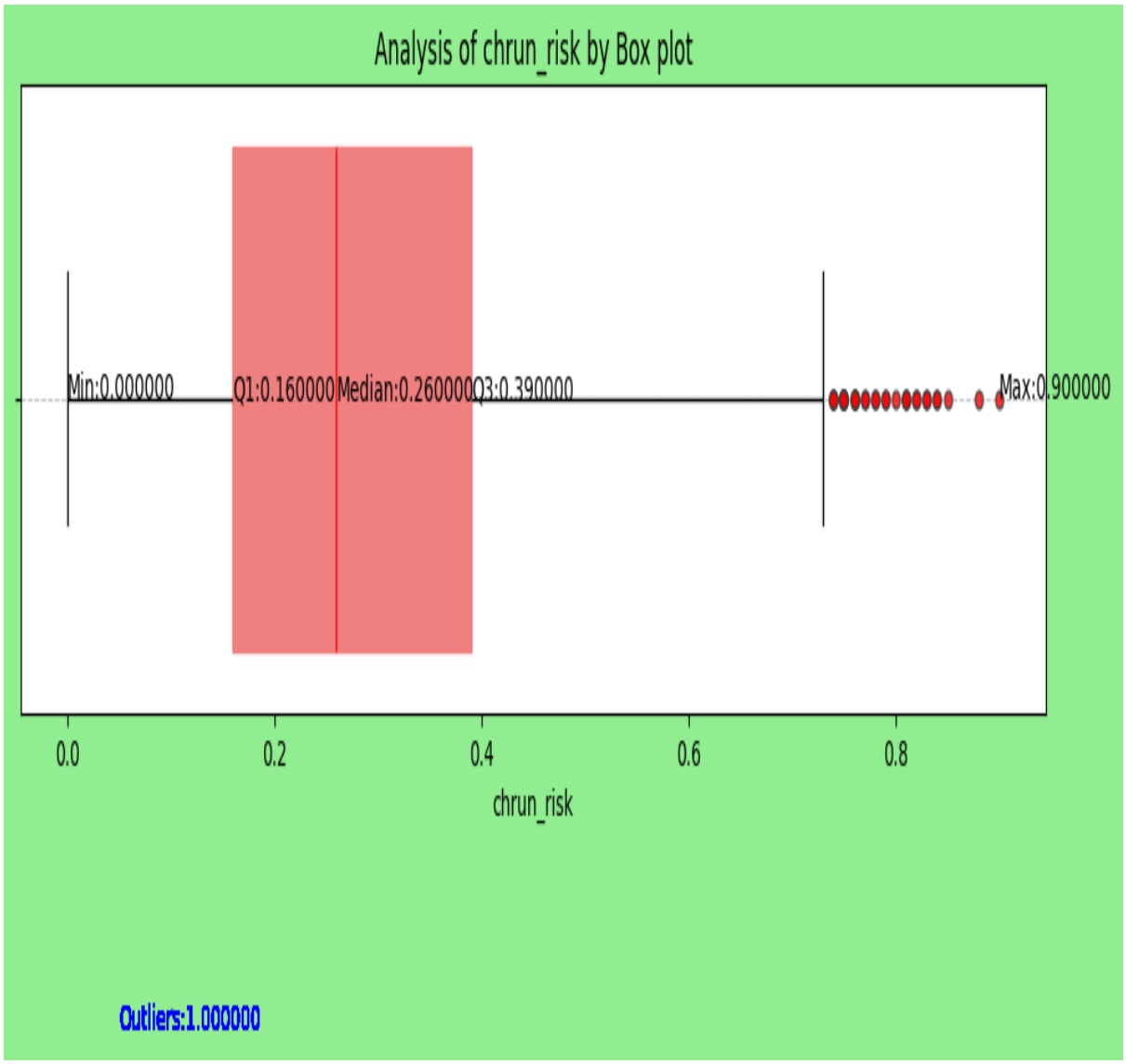
Univariate Analysis

17. Box plot (Analysis of chrun_risk)

Code:

```
# Univariate Analysis
plt.figure(figsize=(10,5),facecolor='lightgreen')
x=sns.boxplot(x=df['churn_risk'],patch_artist=True,
        boxprops=dict(facecolor='c',color='lightcoral'),
        medianprops=dict(color='red'),
        whiskerprops=dict(color='black'),
        capprops=dict(color='black'),
        flierprops=dict(marker='o',markerfacecolor='red',markersize=5,alpha=0.8))
Q1=round(df['churn_risk'].quantile(0.25),2)
Q2=round(df['churn_risk'].quantile(0.50),2)
Q3=round(df['churn_risk'].quantile(0.75),2)
min_val=round(df['churn_risk'].min(),2)
max_val=round(df['churn_risk'].max(),2)
IQR=Q3-Q1
lower=Q1-1.5*IQR
upper=Q3+1.5*IQR
outliers=round(df[(df['churn_risk']<lower)|(df['churn_risk']>upper)]['churn_risk'])
x.text(Q1,0.002,f'Q1:{Q1:2f}')
x.text(Q2,0.002,f'Median:{Q2:2f}')
x.text(Q3,0.002,f'Q3:{Q3:2f}')
x.text(min_val,0.0002,f'Min:{min_val:2f}')
x.text(max_val,0.0002,f'Max:{max_val:2f}')
for val in outliers:
  x.text(0.05,val,f'Outliers:{val:2f}', color='blue')
plt.title('Analysis of chrun_risk by Box plot')
plt.xlabel('chrun_risk')
plt.grid(axis='y',linestyle='--',alpha=0.7)
plt.tight_layout()
plt.show()
```

o/p:


Analysis of chrun_risk by Box plot
Min:0.000000
Q1:0.160000
Median:0.260000
Q3:0.390000
Max:0.900000
0.0
0.2
0.4
0.6
0.8
chrun_risk
Outliers:1.000000

Interpretation of Box plot

- **Title:**

  "Analysis of churn_risk by Box Plot"

- **Explanation:**
  - Above chart is Univariate Analysis.
  - Box plot gives value of Q1, Q2, Q3 minimum and maximum values. Mainly outliers are clearly visible.

- **Saying:**
  - Outliers are present outside of the normal range so here outliers not affecting.

- **Features:**
  - Churn_risk

- **Showing:**
  - Minimum & Maximum values are shown inside the box,25th and 75th percentile values are near the box.
  - Outliers are shown as dots outside the whiskers. so, it does not follow the normal pattern of rest of the data.

Bivariate Analysis

18. Bar Chart (Gender-wise Revenue & Fraud Trend)

Code:

```
avg_rev = df.groupby('Gender')['Revenue'].mean().reset_index()
avg_rev = avg_rev.sort_values(by='Gender', ascending=False)
Is_fraudlent_count=df.groupby('Gender')['Is_fraudulent'].sum()
avg_rev
```

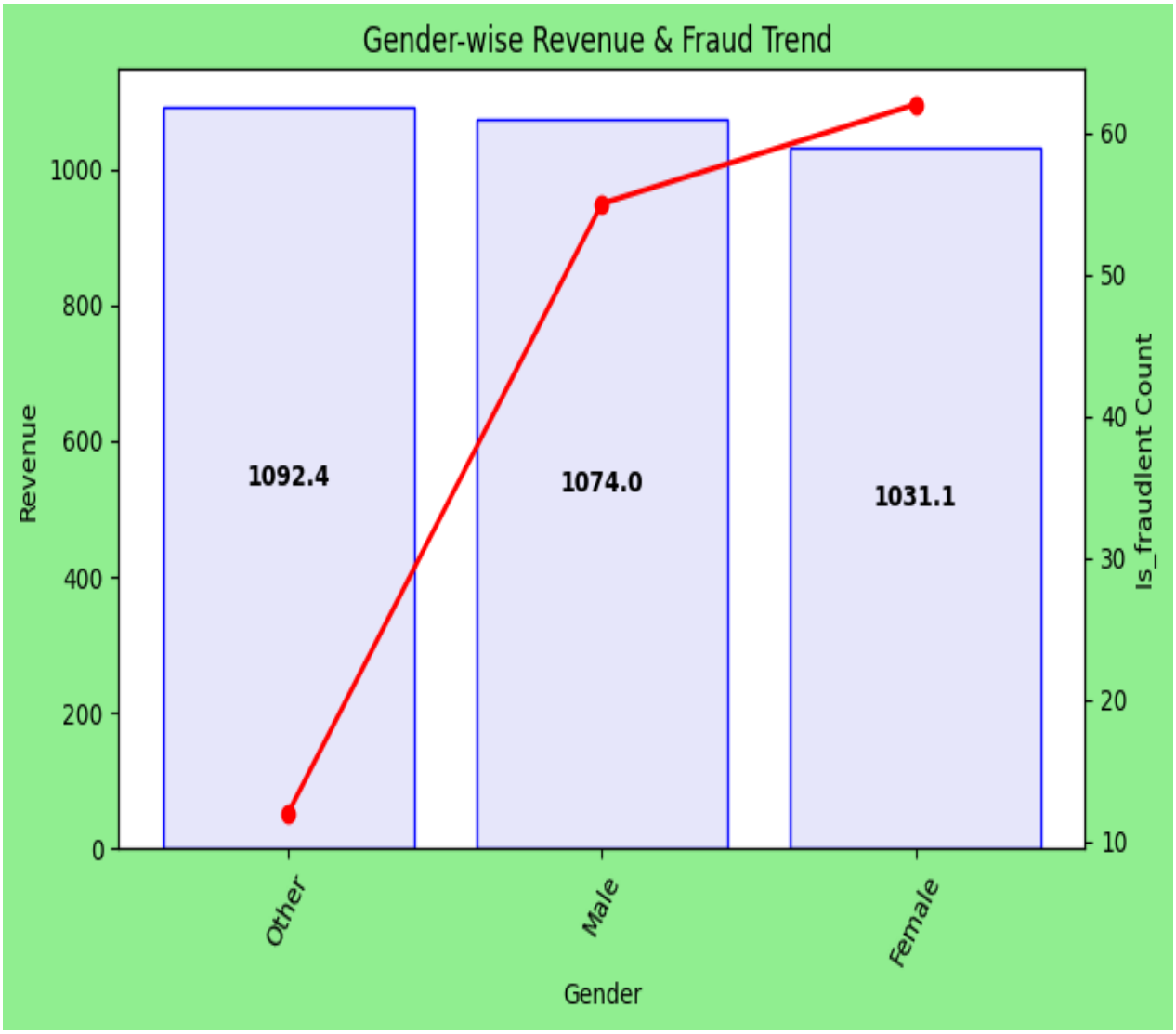
o/p:

	Gender	Revenue
2	Other	1092.350634
1	Male	1074.009148
0	Female	1031.109676

Code:

```
# Bivariate Analysis
plt.figure(figsize=(8,5),facecolor='lightgreen')
bars=plt.bar(avg_rev.Gender,avg_rev.Revenue,color='lavender',edgecolor= 'blue',alpha=1.0)
plt.title('Gender-wise Revenue & Fraud Trend ')
plt.xlabel('Gender')
plt.xticks(rotation=60)
plt.ylabel('Revenue')
for bar in bars:
  height=bar.get_height()
  plt.text(
     bar.get_x()+bar.get_width()/2,height/2,f'{height:.1f}',
     ha='center',va='center',color='black',fontsize=10,fontweight='bold'
  )
x=plt.twinx()
x.plot(Is_fraudlent_count.index,Is_fraudlent_count.values,color='red',marker='o',
                                                                    linewidth = 2)
x.set_ylabel("Is_fraudlent Count")
plt.show()
```

o/p:


Gender-wise Revenue & Fraud Trend
Revenue
Is_fraudlent Count
1092.4
1074.0
1031.1
0
200
400
600
800
1000
10
20
30
40
50
60
Other
Male
Female
Gender

Interpretation of Bar Chart

- **Title:**
  "Gender-wise Revenue & Fraud Trend."

- **Explanation:**
  - Above chart is Bivariate Analysis.
  - Bar chart compares the total revenue by gender category along with the trend of fraudulent transaction counts.

- **Saying:**
  - The amount of revenue each gender generates anf fraud cases vary across.
  - Helps to analyze who contributes more revenue and whether associated with fraud risk.

- **Features:**
  - X-axis = Gender,
  - Y-axis = Revenue,
  - Y-label = Is_fraudulent.

- **Showing:**
  - Bars show Revenue -Others score high, Male scores next, Female scores last.
  - Line trend shows fraudulent count, female category shows the highest fraud count.

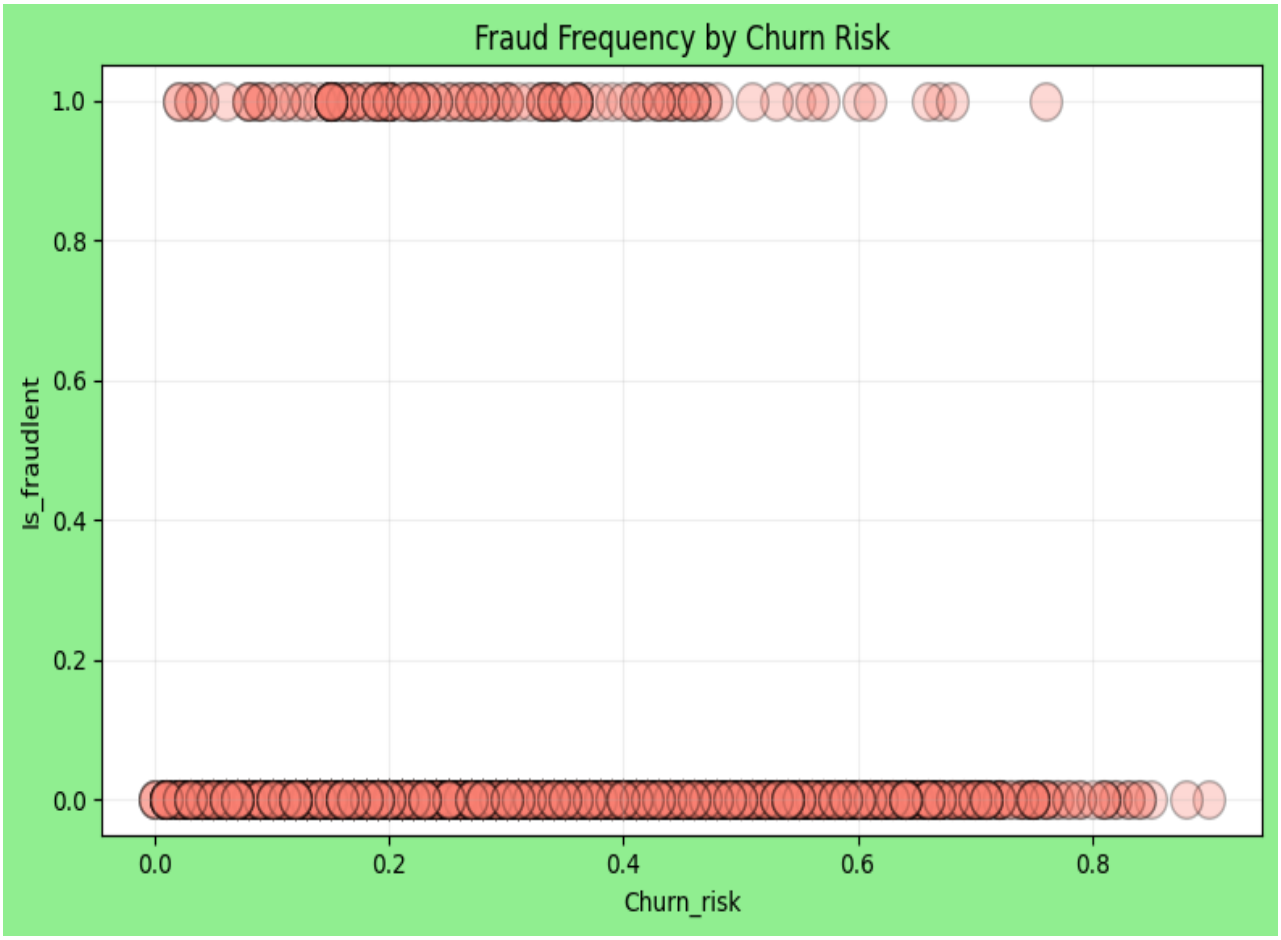
Bivariate Analysis

19. Scatter Plot (Fraud Frequency by Churn Risk)

Code:

```
# Bivariate Analysis
plt.figure(figsize=(8,5),facecolor='lightgreen')
plt.scatter(df.churn_risk,df.Is_fraudulent,s=200,color='salmon',edgecolor=
'black',marker='o',alpha=0.3)
plt.title('Fraud Frequency by Churn Risk')
plt.xlabel('Churn_risk')
plt.ylabel('Is_fraudulent')
plt.grid(alpha=0.2)
plt.tight_layout()
plt.show()
```

o/p:



Interpretation on Scatter Plot

- **Title:**
  "Fraud Frequency by Churn Risk.”

- **Explanation:**
  - Above chart is Bivariate Analysis.
  - The Scatter plot tells fraudulent transactions are distributed across churn-risk.

- **Saying:**
  - Customers with low, medium, high churn risk have mix of fraud and no-fraud.
  - So, fraud is not strongly dependent on churn risk.

- **Features:**
  - X-axis = Churn-risk,
  - Y-axis = Is_fraudulent.

- **Showing:**
  - Points are spread across churn risk but all values fall only on 0 and 1.
  - Because it is binary band of dots.

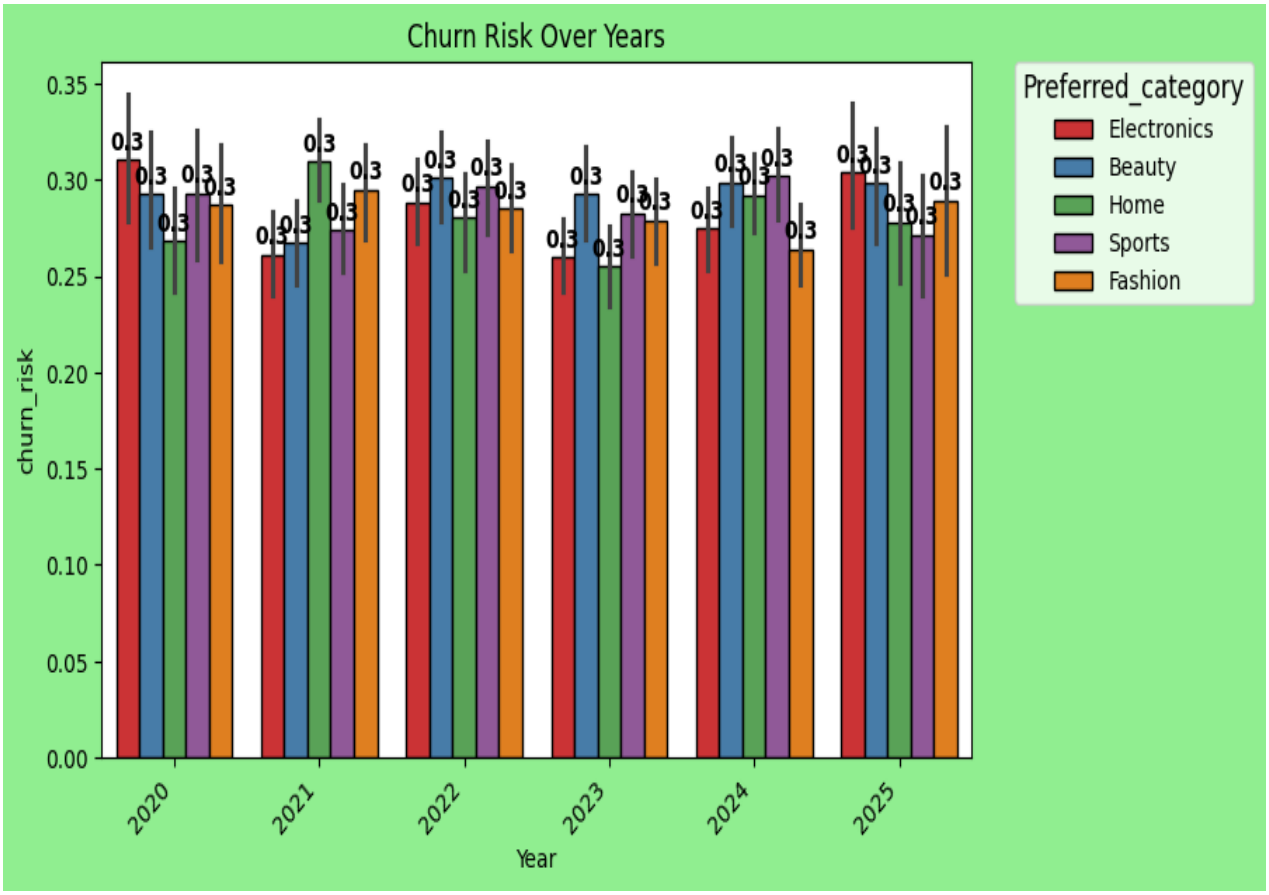
Bivariate Analysis

20-Bar Chart (Churn Risk Over Year)

Code:

```
#Bivariate Analysis
df_sorted = df.sort_values('Year',ascending=False)
plt.figure(figsize=(8,5),facecolor='lightgreen')
x=sns.barplot(data= df_sorted,x='Year',y='churn_risk',
                        hue= 'Preferred_category',palette='Set1',edgecolor='black',alpha=1.0)
plt.title('Churn Risk Over Years')
plt.xlabel('Year')
plt.xticks(rotation=45,ha='right')
plt.ylabel('churn_risk')
for container in x.containers:
  x.bar_label(container,fmt='%.1f',label_type='edge',fontsize=10,fontweight='bold',
                                                        color='black',padding=2)
plt.legend(title='Preferred_category',title_fontsize=12,fontsize=10,
                                bbox_to_anchor=(1.05, 1), loc='upper left',borderaxespad=0)
plt.show()
```

o/p:



Interpretation of Bar chart

- **Title:**
  "Churn Risk Over Year"

- **Explanation:**
  - Above chart is Bivariate Analysis.
  - This above bar chart explains about churn risk among year wise including preferred categories.

- **Saying:**
  - Churn risk stays in the 0.25-0.33 range for almost categories.
  - Categories with taller bars consistently across years indicate higher-risk segments.
  - Some categories slightly fluctuate year to year showing minor variations.

- **Features:**
  - X-axis = Year,
  - Y-axis = churn_risk,
  - hue = Preferred_category.

- **Showing:**
  - Taller bars = higher churn risk.
  - Bars close together show categories have similar values.

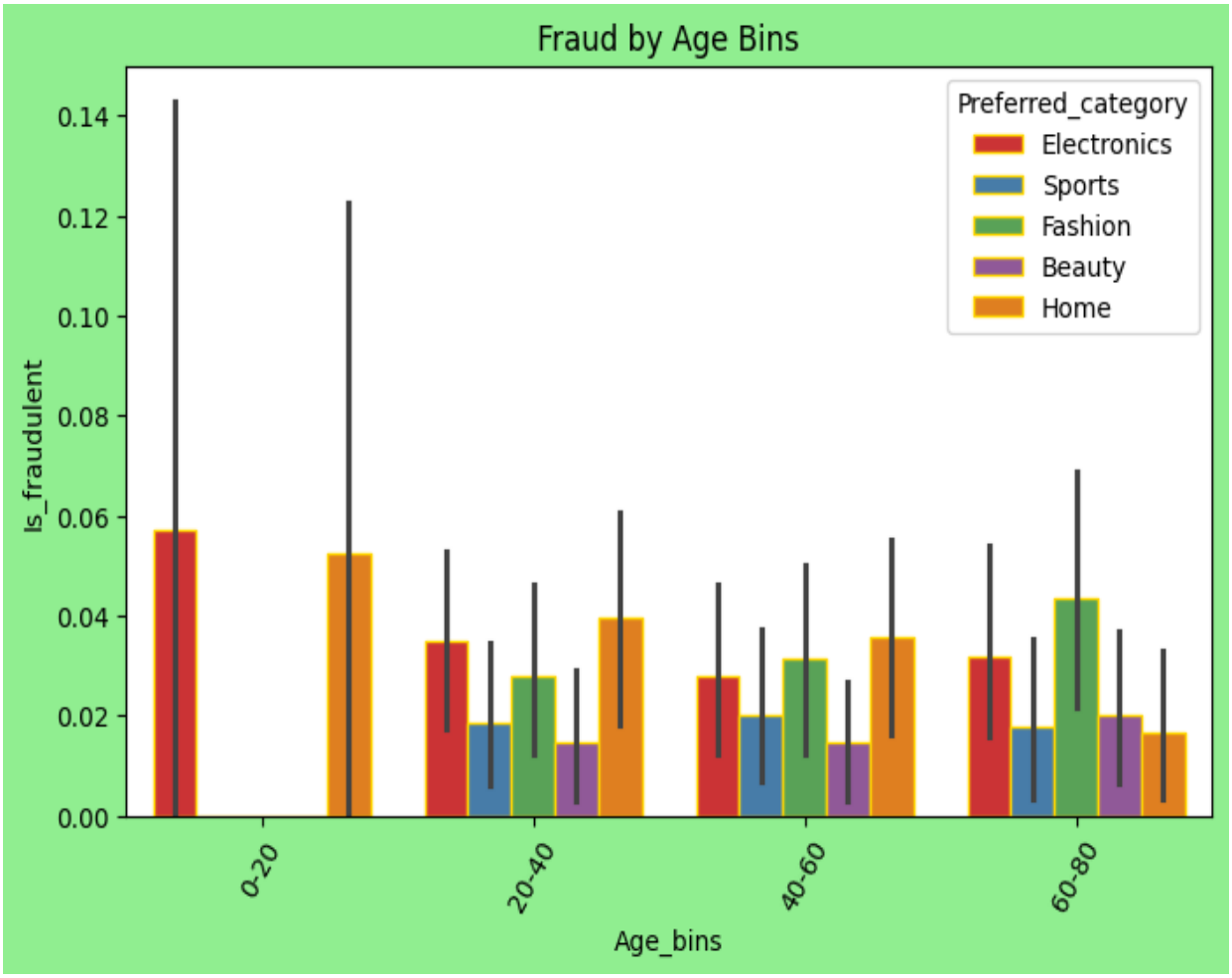
Bivariate Analysis

21. Bar Chart (Fraud by Age Bins)

Code:

```
df.loc[:,'Age_bins']=pd.cut(df['Age'],bins=[0,20,40,60,80],
                                                    labels=['0-20','20-40','40-60','60-80'])
# Bivariate Analysis
plt.figure(figsize=(8,5),facecolor='lightgreen')
x=sns.barplot(data= df,x='Age_bins',y='Is_fraudulent',
                          hue= 'Preferred_category',palette='Set1',edgecolor='gold',alpha=1.0)
plt.title('Fraud by Age Bins')
plt.xlabel('Age_bins')
plt.xticks(rotation=60)
plt.ylabel('Is_fraudulent')
plt.show()
```

o/p:



Interpretation of Bar chart

- **Title:**
  "Fraud by Age Bins."

- **Explanation:**
  - Above chart is Bivariate Analysis.
  - This chart shows how fraud risk varies across different age groups and with changes in each preferred category.

- **Saying:**
  - The 0-20 age group shows highest fraud risk.
  - Fraud risk decreases in the 20-40 and 40-60 age groups, indicating more stable and responsible purchasing pattern.
  - Again, slightly increases in the 60-80 group age customers.

- **Features:**
  - X-axis = Age bins,
  - Y-axis = Is_fraudulent,
  - hue = Preferred_category.

- **Showing:**
  - Each age bin has clustered bars showing category-wise fraud.
  - Taller bars = higher fraud risk, 0-20 group has visibly bars have high fraud.
  - Middle -age group have shorter bars lower fraud risk.

Univariate Analysis

22. Pie Chart (Distribution of Preferred categories)

Code:

```
pc_counts=df['Preferred_category'].value_counts()
pc_counts
```

o/p:

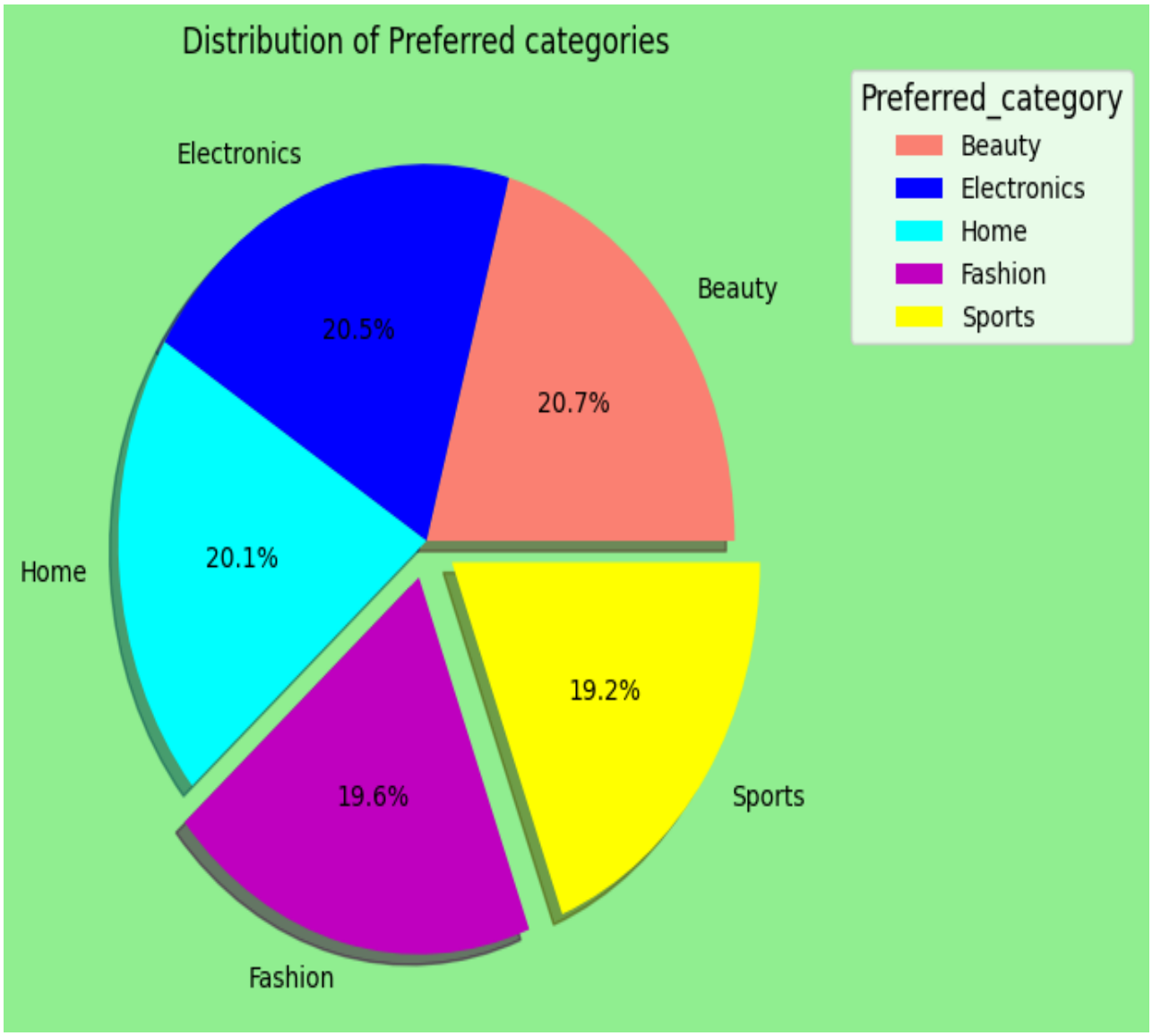
	count
Preferred_category	
Beauty	1024
Electronics	1013
Home	995
Fashion	969
Sports	949

dtype: int64

code:

```
# Univariate Analysis
plt.figure(figsize=(10,6),facecolor='lightgreen')
plt.pie(pc_counts,
      labels=pc_counts.index,
      colors=['salmon','blue','aqua','m','yellow'],
      autopct="%1.1f%%",
       explode=[0,0,0,0.1,0.1],
      shadow=True
      )
plt.title('Distribution of Preferred categories')
plt.legend(title='Preferred_category',title_fontsize=12,fontsize=10,
                                    bbox_to_anchor=(1.05, 1), loc='upper left',borderaxespad=0)
plt.show()
```

o/p:


Distribution of Preferred categories
Preferred_category
Beauty
Electronics
Home
Fashion
Sports
Electronics
Beauty
Home
Sports
Fashion
20.5%
20.7%
20.1%
19.2%
19.6%

Interpretation of Pie Chart

- **Title:**
  "Distribution of Preferred categories."

- **Explanation:**
  - Above chart is Univariate Analysis.
  - Pie chart explains about how preferred categories split among customers.

- **Saying:**
  - Beauty, Electronics and Home are almost divided equally slightly differs among customers.
  - Sports and Fashion are having same values.

- **Features:**
  - Preferred Category.

- **Showing:**
  - Pie charts showing clear visual image about division of preferred categories.

Customer Analytics Loyalty Vs Fraud

DASHBOARD-4

Bivariate Analysis

23. Line chart (Revenue trend of Top-5 countries)

Code:

```
top_5 = df.groupby("Country")['Revenue'].sum().nlargest(5).index
df_top_5 = df[df['Country'].isin(top_5)]
df_top_5
```

o/p:

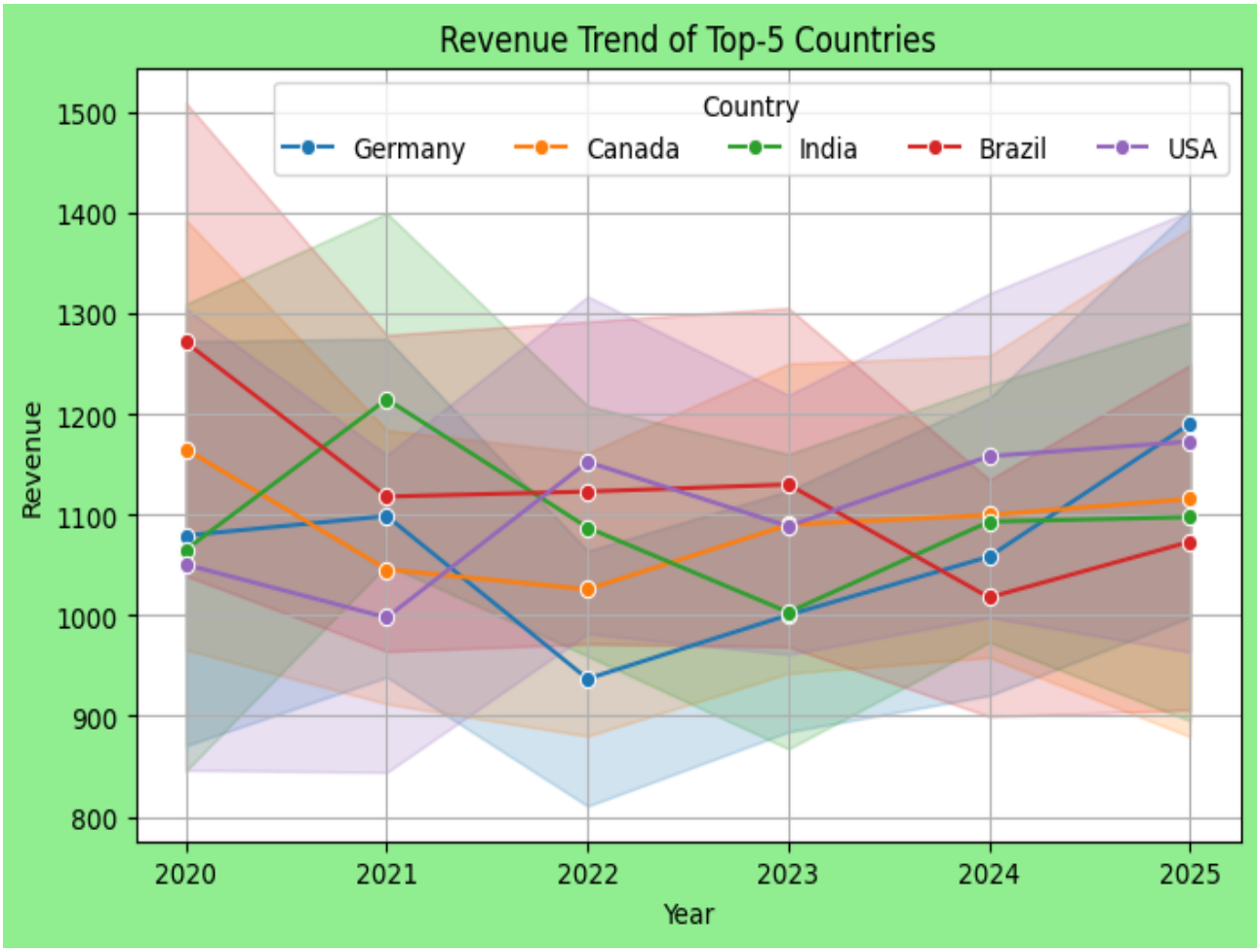
```
top_5 = df.groupby("Country")['Revenue'].sum().nlargest(5).index
df_top_5 = df[df['Country'].isin(top_5)]
df_top_5
```

	CustomerID	Age	Gender	Country	Avgorder_value	Total_orders	Last_purchase	Is_fraudulent	Preferred_cate
0	CUST_9340	43	Female	Germany	10.66	13	346 days	0	Electr
2	CUST_2440	77	Male	Canada	11.48	9	245 days	0	Fas
6	CUST_5563	71	Female	India	14.04	11	47 days		Be
7	CUST_1699	51	Male	Canada	14.16	7	102 days	0	Fas
9	CUST_4192	27	Male	India	14.31	8	100 days	0	S
...	...	...	...	...	...	...	...	...	
4993	CUST_9601	30	Female	Germany	93.19	11	101 days	0	S
4995	CUST_9728	33	Female	Germany	93.19	12	278 days	0	H
4996	CUST_9801	76	Female	Canada	93.19	4	268 days	0	Electr
4997	CUST_9858	23	Female	Germany	93.19	8	193 days	0	Fas
4998	CUST_9970	21	Female	Canada	93.19	9	241 days	0	Electr

2507 rows × 25 columns

```
plt.figure(figsize=(8,5),facecolor='lightgreen')
sns.set_palette("tab10")
sns.lineplot(data=df_top_5,x='Year',y='Revenue',hue='Country',marker='o')
plt.title('Revenue Trend of Top-5 Country')
plt.xlabel('Year')
plt.ylabel('Revenue')
plt.legend(title='Country',ncol=5)
plt.grid(True)
plt.show()
```

o/p:


Revenue Trend of Top-5 Countries
Country
Germany
Canada
India
Brazil
USA
Revenue
1500
1400
1300
1200
1100
1000
900
800
2020
2021
2022
2023
2024
2025
Year

Interpretation of Line Chart

- **Title:**
  "Revenue Trend of Top-5 Countries."

- **Explanation:**
  - Line charts represent yearly revenue of top 5 countries and shaded area help to know revenue fluctuates over time.

- **Saying:**
  - Dots on line highlighting the exact revenue point for each year.
  - Revenue across countries stay mostly in 1000-1400 range.

- **Features:**
  - X-axis = Year,
  - Y-axis = Revenue,
  - hue = Country.

- **Showing:**
  - Most countries show dip in the middle years (2022-2023) and then rise again.
  - Few countries have strong growth from 2024-2025.
  - No country shows straight growth.
  - Many line crossing each other, showing volatility and mixed trends.

Bivariate Analysis

24.Bar Chart (Top-3 Revenue by Preferred Category)

Code:

```
top_3 = df.groupby("Preferred_category")['Revenue'].sum().nlargest(3).index
df_top_3 = df[df['Preferred_category'].isin(top_3)]
df_top_3
```

o/p:

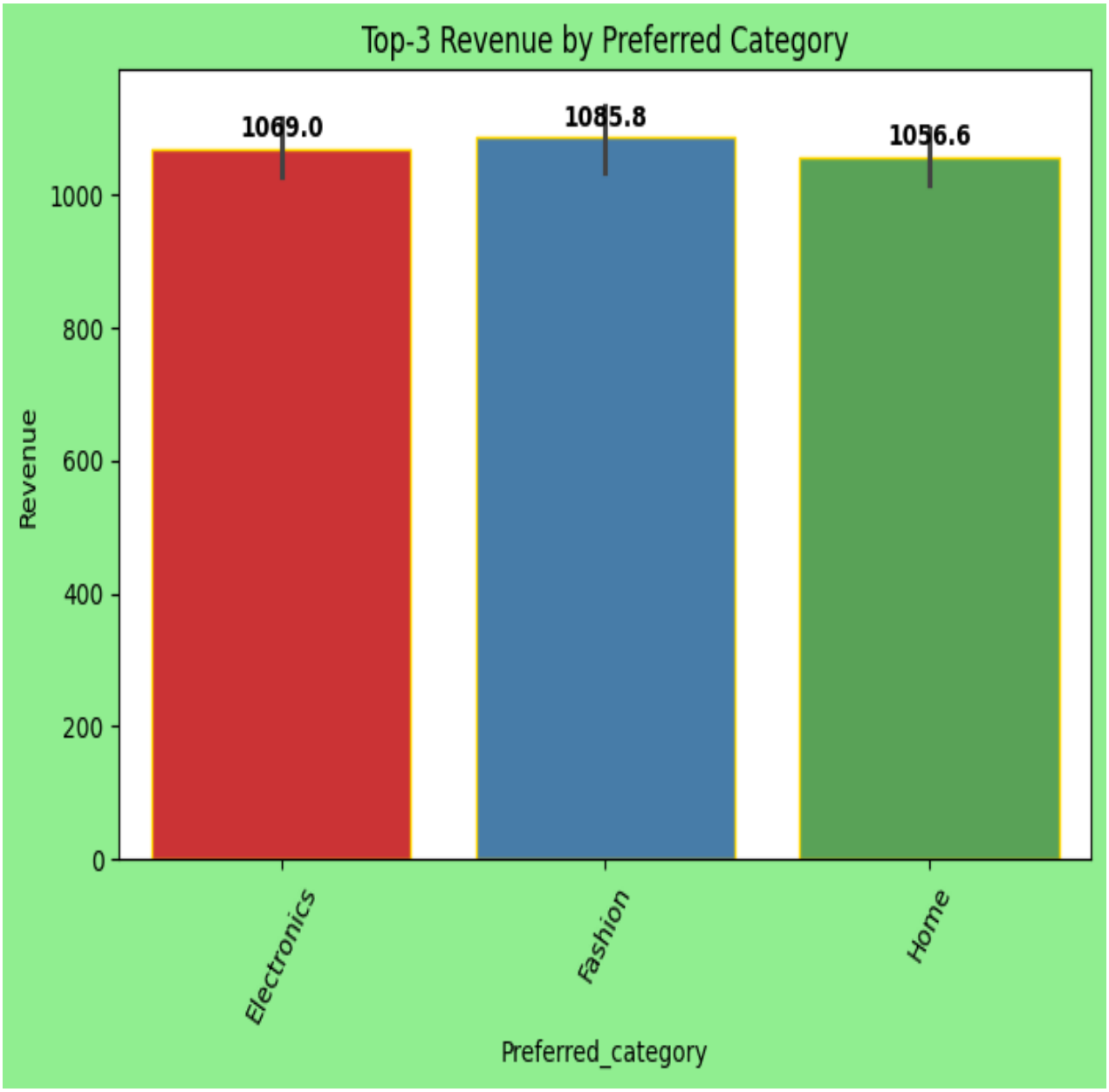
5	CUST_5481	79	Male	China	12.87	8	213 days	0	Fashion	95.9	... Medium|Loy
7	CUST_1699	51	Male	Canada	14.16	7	102 days	0	Fashion	32.8	... Low|Loy
8	CUST_8400	57	Female	Japan	14.19	9	9 days	1	Electronics	73.0	... Low|Frau
...	...	...	...	...	...	...	...	...	...	...	...
4992	CUST_9592	43	Female	Brazil	93.19	14	23 days	0	Home	86.8	... High|Loy
4995	CUST_9728	33	Female	Germany	93.19	12	278 days	0	Home	19.1	... Low|Loy
4996	CUST_9801	76	Female	Canada	93.19	4	268 days	0	Electronics	81.1	... Low|Loy
4997	CUST_9858	23	Female	Germany	93.19	8	193 days	0	Fashion	33.4	... High|Loy
4998	CUST_9970	21	Female	Canada	93.19	9	241 days	0	Electronics	97.3	... High|Loy

2977 rows × 26 columns

Code:

```
#Bivariate Analysis
plt.figure(figsize=(8,5),facecolor='lightgreen')
x=sns.barplot(data=
df_top_3,x='Preferred_category',y='Revenue',palette='Set1',edgecolor='gold',alpha=1.0)
plt.title('Top-3 Revenue by Preferred Category')
plt.xlabel('Preferred_category')
plt.xticks(rotation=60)
for container in x.containers:
  x.bar_label(container,fmt='%.1f',label_type='edge',fontsize=10,fontweight='bold',
                                                          color='black',padding=2)
plt.ylabel('Revenue')
plt.show()
```

o/p:


Top-3 Revenue by Preferred Category
1069.0
1085.8
1056.6
1000
800
600
400
200
0
Revenue
Electronics
Fashion
Home
Preferred_category

Interpretation of Bar chart

- **Title:**

  "Top-3 Revenue by Preferred Category"

- **Explanation:**
  - Above chart is Bivariate Analysis.
  - The value labels on top 3 categories among all.
  - Here Fashion, Electronics and Home are mentioned top 3 categories.
  -
- **Saying:**
  - Each bar represents the total revenue,
    - Fashion= first highest revenue,
    - Electronics = second highest revenue,
    - Home= third highest revenue.

- **Features:**
  - X-axis = Preferred_category,
  - Y-axis =Revenue.

- **Showing:**
  - The bars are tall and close in height, meaning all categories are performing similarly with small differences.
  - The labels above the bar make comparison exact without needing to estimate from the axis.

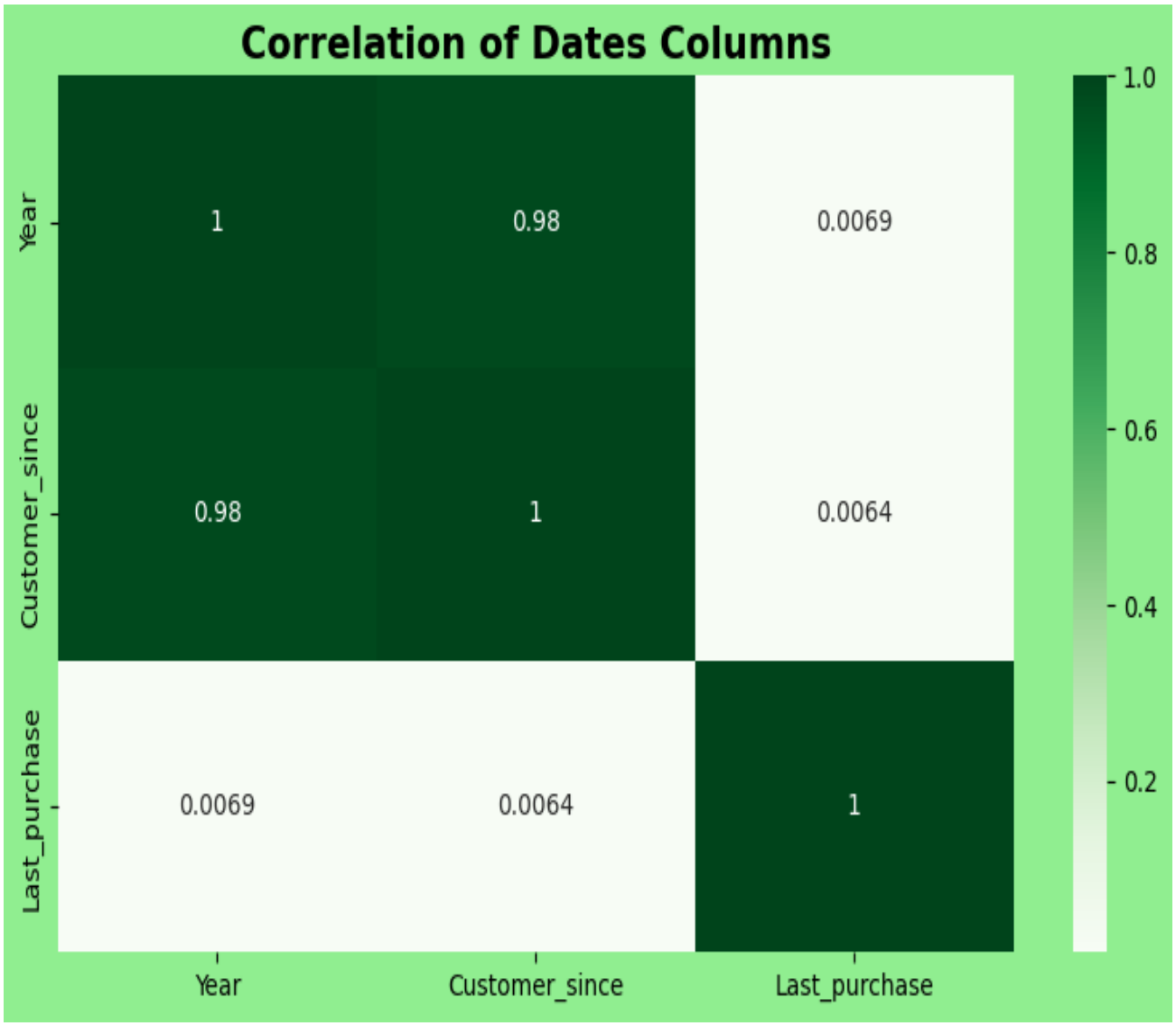
Multivariate Analysis

25. Heatmap (Correlation of Dates Columns)

Code:

```
# Multivariate Analysis
matrix = df[['Year','Customer_since','Last_purchase']].corr()
plt.figure(figsize=(8,5))
sns.heatmap(matrix,annot=True,cmap= 'Greens')
plt.title('Correlation of Dates Columns',fontweight='bold',fontsize=16)
plt.tight_layout()
plt.show()
```

o/p:



Interpretation of Heatmap

- **Title:**
  "Correlation of Dates Columns."

- **Explanation:**
  - Above chart is Multivariate Analysis.
  - Heatmap is the correlation between different columns and number inside tells strongly how two variables are connected.

- **Saying:**
  - Year and Customer_since have very strong Positive correlation.
  - Year and Last_purchase have almost zero correlation.
  - Customer_since and Last_purchase also have almost zero correlation.

- **Features:**
  - Year, Customer_since, Last_purchase.

- **Showing:**
  - Shows all diagonal values are 1 which means every variable is perfectly correlated with itself.

Multivariate Analysis

26.Pivot Table (Month-wise Revenue)

Code:

```
# MultiVariate Analysis
df.loc[:,'Month']=df['Customer_since'].dt.month
pivot= df.pivot_table(
   values='Revenue',
   index='Year',
   columns='Month',
   aggfunc='mean'
)
pivot
```

o/p:

Month	1	2	3	4	5	6	7	8	9	10	11	12
Year												
2020	NaN	NaN	NaN	NaN	NaN	946.98	1144.36	1044.81	1049.51	1263.73	901.92	994.30
2021	1126.45	975.58	1202.99	1084.66	1008.19	1228.86	963.06	1023.69	1062.60	1022.21	1048.13	1006.03
2022	951.96	1061.86	1018.15	1035.47	1041.48	1020.59	951.45	921.00	1102.51	1048.43	1086.10	1032.89
2023	1069.57	911.06	1031.31	1180.55	1063.31	1163.39	958.45	1103.27	1109.34	1014.17	972.91	1039.68
2024	1058.89	1135.98	1096.49	1082.51	1107.54	1018.37	966.90	1063.58	1135.82	947.98	1090.33	945.23
2025	1043.87	1167.48	1102.42	1204.32	1092.96	1024.97	NaN	NaN	NaN	NaN	NaN	NaN

Interpretation of Pivot Table

- **Title:**
  "Month-wise Revenue."

- **Explanation:**
  - Above chart is Multivariate Analysis.
  - Pivot table is plotted for different Years according to their months comparing with revenue amount

- **Saying:**
  - Monthly revenue value based on the years.

- **Features:**
  - Revenue, Year, Customer_since, Month

- **Showing:**
  - 2021,2022,2023 & 2024 having high revenue.
  - 2020 & 2025 have less revenue value because dataset year starts on June 2020 and ends on June 2025.
  - NaN also shows because of this reason.

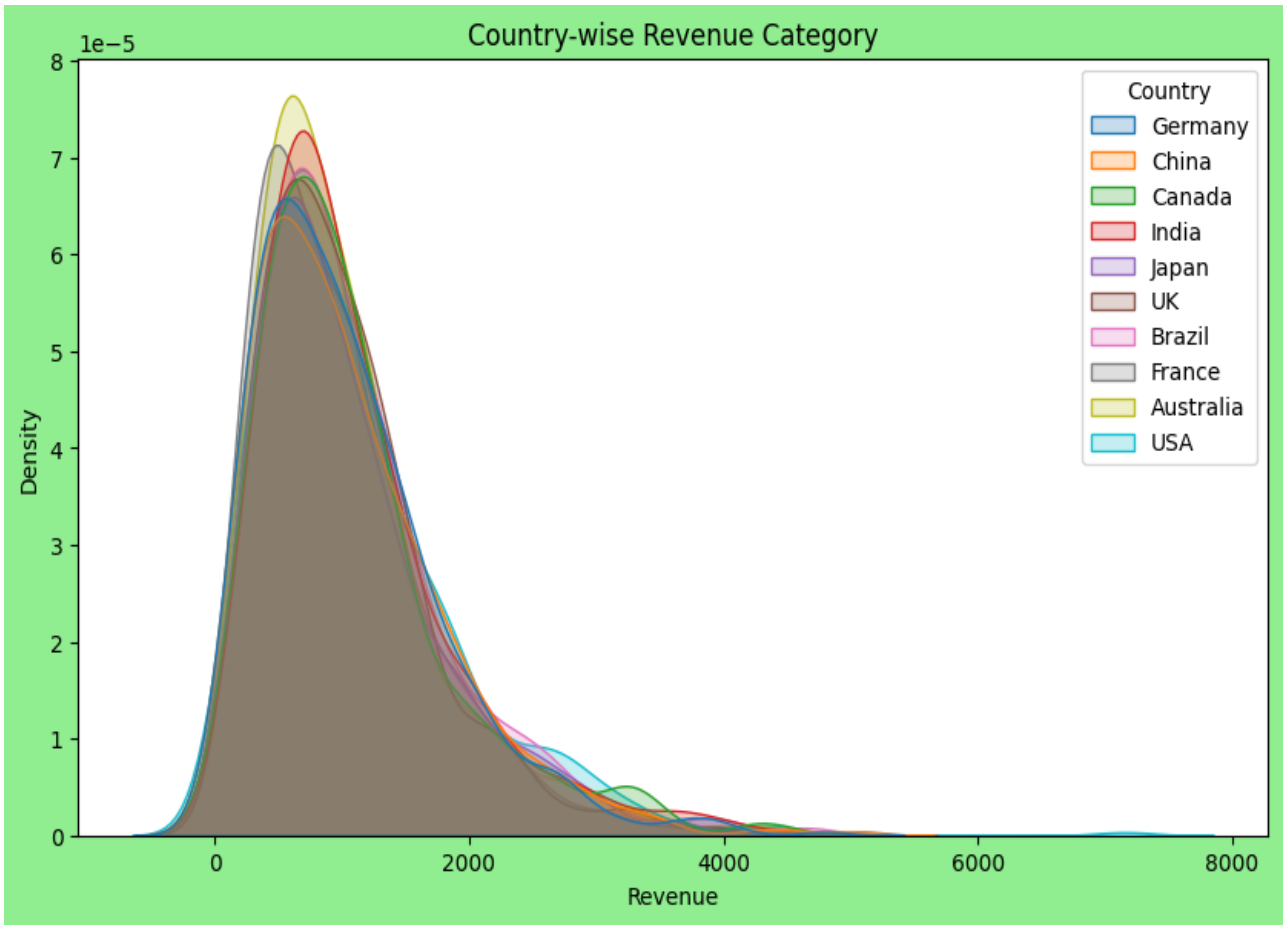
#kde density plot - Univariate Analysis

27. Density plot (Country-wise Revenue Category)

Code:

```
#kde density plot - Univariate Analysis
plt.figure(figsize=(10,6),facecolor='lightgreen')
sns.kdeplot(
   x='Revenue',
   hue='Country',
   fill=True,
   data=df,
   bw_adjust=1
)
plt.title('Country-wise Revenue Category')
plt.show()
```

o/p:



Interpretation of Density Plot

- **Title:**
  "Country-wise Revenue Category."

- **Explanation:**
  - Above chart is Univariate Analysis.
  - The kde plot helps identify countries have higher revenue and they spread. Countries with broader peaks
  - Countries with broader peaks shows more variability, while sharper peaks show stable.

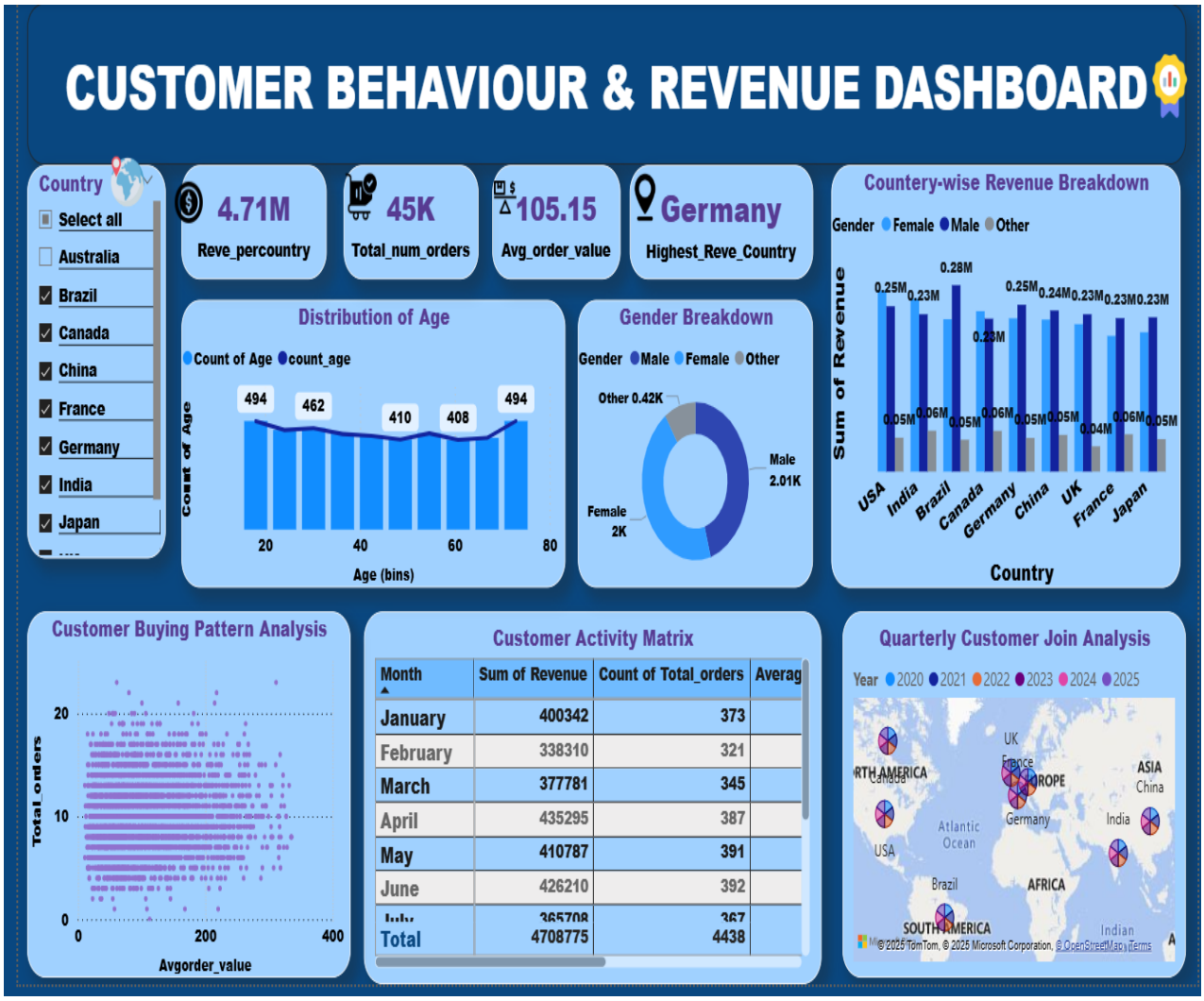
- **Saying:**
  - By comparing curves, identify which country can contribute more revenue.

- **Features:**
  - X-axis = Revenue,
  - hue = Country.

- **Showing:**
  - Highest curve shows most concentrated revenue.
  - Peak shows customer falls revenue.
  - Wider curve shows greater revenue variability.
  - If curve overlap shows both countries have similar revenue distribution.

Stage 4 – Documentation, Insights and Presentation

DASHBOARD-1



Interpretation of Dashboard 1

Title:

"CUSTOMER BEHAVIOUR & REVENUE DASHBOARD"

Explanation:

- This dashboard explains about the customer interact with the revenue.
- It combines demographic insights, purchase patterns, product preferences and geographic distributions of performance.

Customer Analytics Loyalty Vs Fraud

KPI Highlights:

- Total revenue = shows overall income.
- Total orders = shows total transactions placed.
- Top country = highest revenue collected country.
- Revenue per country = amount collected by each country.

Silicer:

- This silicer lists all customer countries (Australia, Brazil, Canada, China, France, Germany, India, Japan, UK, USA).
- To filter the entire dashboard based on selected country.

Customer Demographic Analysis:

- Age distribution chart = shows about customers are spreads across age group.
- Gender distribution Donut chart = understanding the gender contributes.

Buying Pattern Analysis:

- Bar charts display most purchased product categories of customer preferences.
- Average revenue by category shows the highest earning per customer.

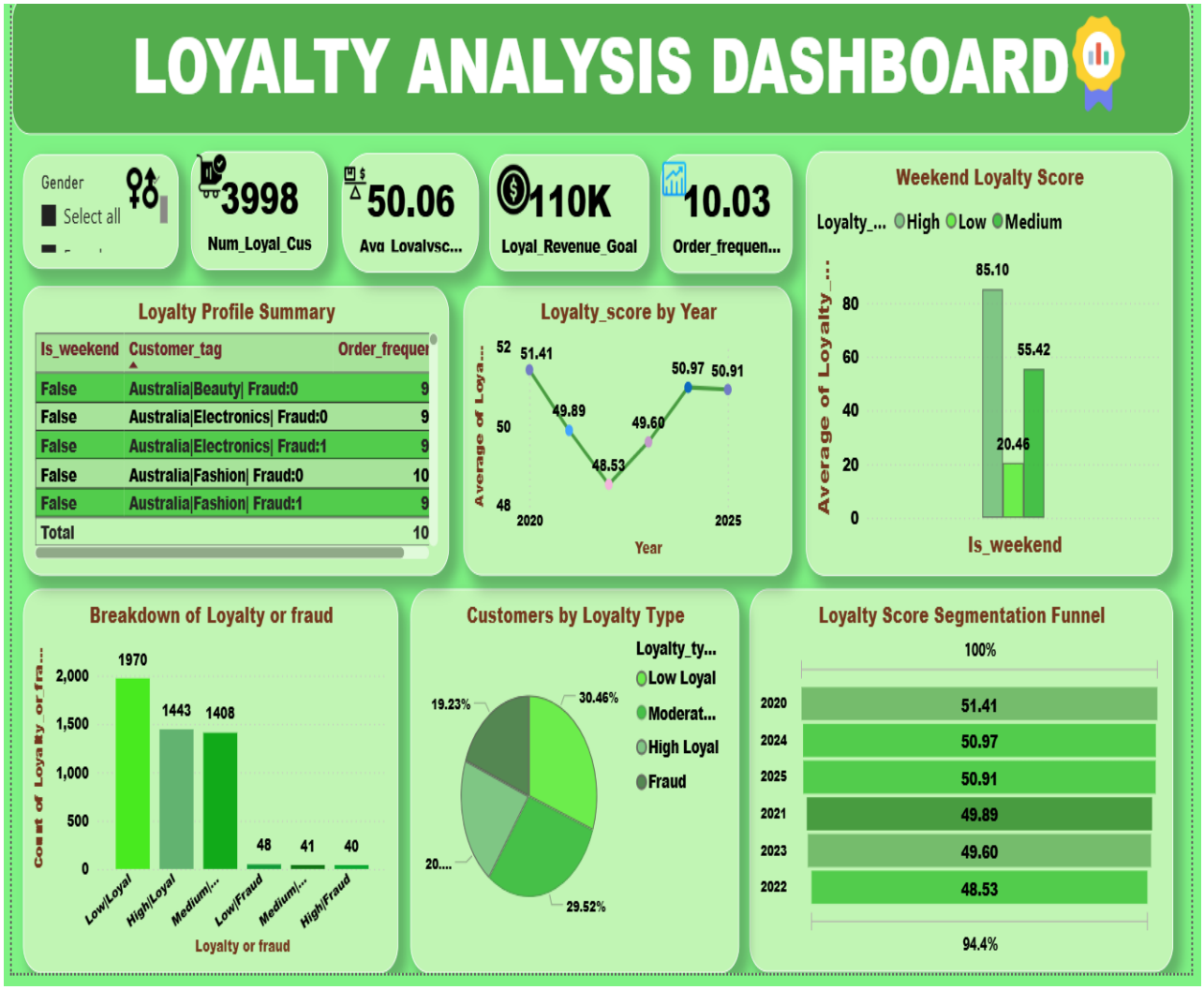
Country-wise Customer Spread:

- A Map visual shows the customer exact location based on high performance regions.
- This supports geo-targeted marketing and regional strategy planning.

Customer Activity Metrics:

- Displays the visit frequency, purchasing frequency, repeat vs one-time customers.

DASHBOARD-2



Interpretation of Dashboard 2

Title:

"LOYALTY ANALYSIS DASHBOARD"

Explanation:

- This dashboard gives a loyal customer engagement and revenue values.

Customer Analytics Loyalty Vs Fraud

KPI Cards:

- Number of Loyal Customers = gives exact count,
- Average Loyalty Score = gives average loyalty score value,
- Loyal Revenue Goal = It gives the goal amount,
- Order Frequency = gives the approximate rate of order frequently customers.

Silicer:

- Here silicer is gender Male, Female and Others.
- To filter the entire dashboard based on selected gender.

Loyalty Distributions:

- A Pie chart shows the customers are spread across loyalty type.
- Weekend loyalty score gives information of revenue collected during weekends.
- Loyalty customer segmentation created by using funnel chart.

Behavioural patterns:

- A bar chart gives the information about the loyal or fraud customers based on loyalty pattern.

DASHBOARD-3

Interpretation of Dashboard-3

Title:

"FRAUD & RISK CUSTOMER DASHBOARD"

Explanation:

- This dashboard gives complete overview of fraudulent customer behaviour.

Customer Analytics Loyalty Vs Fraud

KPI Cards:

- Fraud Count = total number of fraudulent customers,
- Fraud Revenue = total revenue generated from the fraudulent customers,
- Avg Churn Risk = shows the average churn probability.

Silicer:

- Here silicer is Is_fraudulent whether true or false.
- To filter the entire dashboard based on selected silicer ‘0’ or ‘1’.

Bar Chart:

- Average of churn risk changes over years, horizontal bars make year to year comparison.

Clustered Column Chart:

- A visual compares of average revenue and fraud count by genders.

Tree Map:

- Shows fraud count across countries and find high-risk countries.
- Countries with lower fraud involvement.
- Regional fraud patterns.

Donut chart:

- Shows the high risk, medium risk and low risk customers.

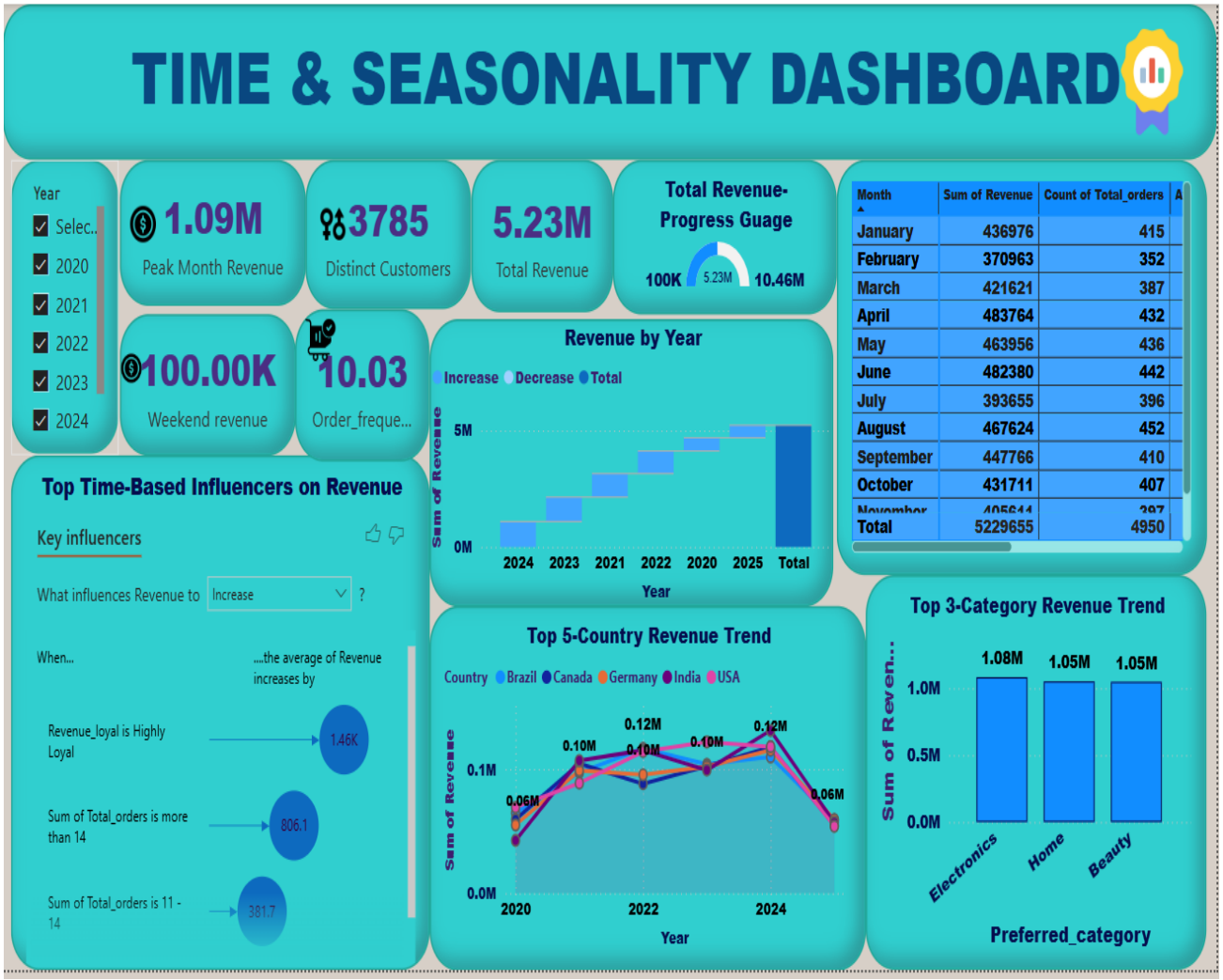
Stacked Column Chart:

- This chart shows a fraud varies by age groups and high-risk customers age bins.

Scatter Plot:

- This chart helps to find frequently customers commit fraud.

DASHBOARD-4



Interpretation of Dashboard-4

Title:

"TIME & SEASONALITY DASHBOARD"

Explanation:

- This dashboard shows revenue, orders, customer behaviour and demand patterns change over time.
- It highlights the yearly performance and top five ratings for clear understanding.

Customer Analytics Loyalty Vs Fraud

KPI Cards:

- Peak Month Revenue = strongest month revenue,
- Distinct Customer = Unique customer purchased,
- Total Revenue = Overall amount generated,
- Weekend Revenue = shows Saturday and Sunday revenue,
- Order Frequency = average number of orders per customers.

Silicer:

- Here silicer is year based from 2020 to 2025.
- To filter the entire dashboard based on selected year.

Guage:

- Guage compares current revenue with the target range and how much achieved.

Month-wise Revenue & Order Table:

- This table shows the month wise revenue, total number of orders, Average order values.
- Strong month =March, April July
- Low periods = January, February.

Bar Chart:

- Gives trend growth and peak performing years and revenue year to years.

Key Influencer:

- Explains time-based factors increase or decreases revenue.
- More order = revenue increases
- High Loyalty score = revenue increases
- Seasonal month = increase or decrease revenue

Line chart (Top-5):

- Shows the top 5 countries geographic performance of revenue over years.

Line chart (Top-3):

- Shows the top 3 categories of high demand seasons and revenue cycles.

Future Enhancement

- Advanced Predictive Analytics
  - Forecast customer churn
  - Predict future revenue
  - Fraud probability prediction

- Automated Alerts & Notifications
  - Sudden drop in revenue
  - High churn risk customers
  - Weekend dips or seasonal variations

- Drill-Through & Deep-Dive Pages
  - Country-wise deep revenue insights
  - Month-wise and week-wise patterns

- AI powered Insights
  - AI based suggestions
  - Automatic explanation of trends

- Segmentation and Clustering
  - Age groups
  - Risk groups
  - Geography clusters

- Recommendation Systems
  - Best-product category to promote this month
  - Which regions need fraud monitoring
  - Which month need marketing push

Conclusion:

- Above Dashboards gives a complete actionable view of customer patterns, revenue performance, risk levels, and time-based trends.
- Each dashboard converts raw data into meaningful insights about customers loyalty and revenue details.
- Supports better decision- making in areas like customer retention, revenue planning, marketing strategy.
- Over-all this system delivers a data-driven foundation that helps organization to grow revenue, reduce risk, strength customer loyalty and plan future strategies with confidence.

Thank you